Master’s Thesis

Master's Degree in Industrial Engineering

What is the challenge in creating a process-based digital twin?

Fachgebiet Logistik
Technische Universität Berlin

Professor:
Prof. Dr.-Ing. Frank Straube

Supervisor:
Tino T. Herden, M.Sc.

Student:
Cristina Rodríguez Murciano
Student Nr. 0407815

Delivery: July 2019

Abstract

Supply chain management (SCM) has become increasingly relevant for organisations. Firms that have a strategy to optimise their performance in the supply chain (SC) are more prone to be successful. Most companies use data analytics for this purpose, and in order to continuously look for competitive advantage. One of the most important issues that organisations must deal with in SCM is inventory control. This thesis proposes the digital twin technology to solve this problem, by means of predictions made from historical data. This thesis conducted a series of simulation experiments to test the capacity of a digital twin simulation to be a better decision-maker than a classical model. The classical model chosen as a baseline was the economic order quantity (EOQ) model. The digital twin was designed with a reinforcement learning (RL) application based on a neural network (NN) and trained several times. Three different trials challenging the limitations of the baseline model were carried out. In order to overcome the EOQ model limitations, two delivery reliability indicators were created, thus allowing to generate different scenarios. Results showed that the inventory level and costs are affected in a different way depending on which reliability parameter is modified. The digital twin did not beat the EOQ model in any of the trials but an approach to it was achieved. Although the same number of iteration trainings were run in all the trials, the learning level reached was not the same. In two of the three trials, 30% of the experiments led to the same results as the classical model, whereas in the last one only 10% of the experiments reached them. This study is only a first approach to a big issue, the SCM. The digital twin can consider other external factors that classical models cannot. However, lots of resources that were not available in this project would be needed in order to properly model and simulate a real-world situation.

Keywords: supply chain management, digital twin, neural network, reinforcement learning.

Abstract

Supply chain management (SCM) wird für Unternehmen immer relevanter. Unternehmen, welche eine Strategie für die Optimierung der Lieferkette haben, neigen mehr dazu erfolgreich zu sein. Die meisten Firmen setzten zu diesem Zweck Datenanalysen ein, um dauerhaft Wettbewerbsvorteile zu erreichen. Einer der wichtigsten Aspekte, mit denen sich Firmen in SCM befassen müssen, ist die Bestandskontrolle. In dieser Arbeit wird die Digital-Twin-Technologie vorgeschlagen, um dieses Problem mithilfe von Vorhersagen aus historischen Daten zu lösen. Es wurde eine Reihe von Simulationsexperimenten durchgeführt, um die Fähigkeit der Digital-Twin-Simulation zu testen, ein besserer Entscheidungsträger als ein klassisches Model zu sein. Als klassisches Basismodel wurde das economic order quantity (EOQ) Modell ausgewählt. Der Digital-Twin wurde mit einer neural network (NN) basierenden reinforcement learning (RL) Anwendung entwickelt und mehrmals trainiert. Es wurden drei verschiedene Testläufe durchgeführt, um die Einschränkungen des Basismodels zu hinterfragen. Um die Einschränkungen des EOQ-Modells zu überwinden, wurden zwei Indikatoren für die Lieferzuverlässigkeit erstellt, mit denen verschiedene Szenarien generiert werden können. Die Ergebnisse zeigten, dass der Lagerbestand und die Kosten in Abhängig davon, welcher Zuverlässigkeitsparameter geändert wird, unterschiedlich beeinflusst werden. Der Digital-Twin hat das EOQ-Modell in keinem der Versuche geschlagen, es wurde jedoch ein Ansatz dafür gefunden. Obwohl in allen Versuchen die gleiche Anzahl von Wiederholungstrainings durchgeführt wurde, war das erreichte Lernniveau nicht das gleiche. In zwei der drei Testläufe führten 30% der Versuche zu den gleichen Ergebnissen wie das klassische Modell, während im letzten Versuch nur 10% der Versuche diese erreichten. Diese Studie ist nur ein erster Ansatz für das große Thema SCM. Der Digital-Twin kann andere externe Faktoren berücksichtigen, welche klassische Modelle nicht berücksichtigen können. Es wären jedoch viele Ressourcen erforderlich, die in diesem Projekt nicht verfügbar waren, um eine reale Situation passend zu modellieren und zu simulieren.

Schlagwörter: supply chain management, digital twin, neural network, reinforcement learning

Statutory declaration

I herewith formally declare that I have written the submitted thesis independently. I did not use any outside support except for the quoted literature and other sources mentioned in the paper. I clearly marked and separately listed all of the literature and all of the other sources which I employed when producing this academic work, either literally or in content.

Cristina Rodríguez Murciano

Berlin, July 2019

Table of content

List of figures

List of tables

Abbreviations

AI	Artificial Intelligence
ANN	Artificial Neural Network
DC	Distribution Center
EOQ	Economic Order Quantity
IoT	Internet of Things
LSCM	Logistics and Supply Chain Management
ML	Machine Learning
RL	Reinforcement Learning
SC	Supply Chain
SCM	Supply Chain Management

1. Introduction

1.1. Motivation

Most companies, independently from size, are involved in one or several supply chains. A supply chain (SC) can be as simple as a supplier providing materials to a factory and then the products obtained from the factory directly sold to retail stores, or it can consist of an enormous network of processes. Nobody knows better than the firms themselves how the supply chain works and how their activities can affect the performance of the whole process. Consequently, it is essential to put the necessary attention on managing supply chains.

When studying a company's performance and its contribution to the supply chain, it is important to get to know first some aspects, such as how the production process works, the company's business model, and which are its goals and the scope of its activity. The question is whether these aspects are enough to properly manage a supply chain.

Other factors that could also influence the supply chain behaviour, and would be therefore interesting to study, are the external factors that can have an indirect contribution to the process or to the strengths and weaknesses of the company. For example, a company's performance could be affected by lead time variability, demand variability and delivery quantity or quality reliability, among others. This means that the same company could be exposed to different varying scenarios that should be considered when making decisions.

Nowadays, data and performance indicators have become usual in supply chains, and the use of analytics has proven to have positive effects on logistics performance (Trkman *et al.*, 2010; Wang *et al.*, 2016). However, not all companies are convinced about usefulness of analytics in logistics, although users emphasize the benefits (Schoenherr and Speier-Pero, 2015).

According to Deloitte's Supply Chain Analytics Guide in 2012, supply chains are a rich place to look for competitive advantage, because of the important role they play in the cost structure of the firm. Using analytics in supply chains gives the company a tool to improve its supply chain in a way that it was not possible in the past. Basing the actions to perform in a process in order to improve it only on past demand and the other factors commented above could possibly imply that other big opportunities could be missed. Making decisions based only on what happened in the past may no longer provide a competitive advantage (Deloitte, 2012).

In this way, it is important to work on a strategy to optimise the performance of supply chains including the interpretation of the data gathered and taking into account the forecasts made from it. This goes through the digitalization of supply chains. The article *Predictions for worldwide Supply Chains in 2019* by Simon Ellis helps to understand the importance of digitally enable companies' supply chains (Ellis, 2019):

> *"By 2021, smart supplier lifecycle management solutions will automate 50% of suppliers' enterprise activities, from onboarding to exit, thus improving both performance and relationships."*

> *"By 2022, over 40% of manufacturers worldwide will be integrating data from product lifecycle apps into their supply chain data to improve overall after-sales service levels, achieving increases of 60%."*

> *"By 2023, talent shortages in the supply chain for 75% of the top 500 manufacturers worldwide will largely have been mitigated by the use of supply chain digital assistants."*

Evidence shows that digital transformation is moving forward and that companies should adapt to it if they want to succeed. These digital changes will require an effort and probably significant changes in operational, tactical and strategical operations.

1.2. Structure

This thesis is divided into different chapters. Chapter 1 deals with the motivation and introduction of the topic as well as with the research question, while Chapter 2 and Chapter 3 present the theory. Chapter 2 introduces the concepts of supply chains, logistics and supply chain management. It also discusses the importance of data and analytics in supply chain management. Chapter 3 describes what is a digital twin and how does it work, and two different kinds of digital twin are presented: object-based digital twin process-based digital twin. It also glances at the current situation of this technology and its future expectations.

Chapter 4 is based on the methodology. First, the model chosen as a baseline, along with its assumptions and limitations is detailed. After that, the concepts of neural network and reinforcement learning are presented. Finally, this chapter is focused on introducing the experimental setup. Different factor combinations to test are raised.

Chapter 5 focuses on the results. First, the expected results are exposed, and subsequently, the experimental results and its implications are discussed. Finally, Chapter 6 consists of the conclusion, including a summary, limitations and future research on the topic.

1.3. Research questions

As mentioned before, a supply chain can take different shapes and have different degrees of complexity. In Figure 1 it is shown a generic supply chain consisting of factories, warehouses and retails. In this particular SC, three factories supply to another factory and warehouse, which in turn supply to another warehouse and retails, and finally to a last retail.

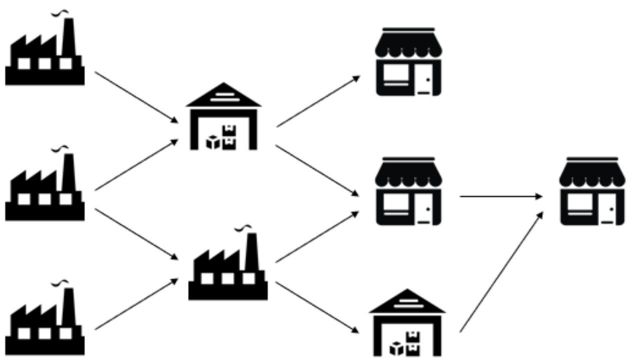


Figure 1. Generic Supply Chain

Each one of the SC makes its own decisions which affect not only to their own performance and profit but also to the performance and profit of the other entities. For example, if one of the factories in the first level of the SC has a delay on the delivery to the factory in the second level, not only the second factory will be resented but also the warehouse in the third level and the retails in both third and fourth level. With this example, it is easy to understand that the performance of the whole supply chain is due to the activities carried out by each of the individuals taking part in the process. Besides, even if each entity has its own decision maker, they will be always affected by the uncertainty of the other entities' performance.

The idea of this is to focus on one of the individuals of the supply chain and test whether a digital twin, after being trained, can be able to be a better decision maker than a classical model taken as a baseline. In this case, because of its simplicity, the Economic Order Quantity model is chosen as a baseline. Thus, the main research question is the following one:

What is the challenge in creating a process-based digital twin?

Besides, other questions emerge from this one:

- Are digital twins useful solutions for companies?
- Which factors should be used for the comparison with a classical model?
- Does the digital twin always beat the classical model?

The aim of this thesis is to answer all the previous questions in order to conclude whether the implementation of digital twins in supply chains would a suitable solution for companies and would improve their performance, and therefore the performance of the whole supply chain.

2. Logistics and Supply Chain Management

This chapter introduces the concepts of supply chain (SC), logistics and supply chain management (SCM). The first two terms are defined, and the concept of SCM is widely described. Its description includes the definition itself, and the objectives and scope, activities, and uncertainties of the SCM. Besides, the importance of data and analytics in SCM is exposed.

The three concepts have been discussed extensively by several authors and there is a certain disagreement regarding the definitions. The definitions used in this thesis are the ones presented below.

2.1. Definition of supply chain and logistics

According to the article *Defining Supply Chain Management of the Journal of Business Logistics*, a supply chain is defined as *"a set of three or more entities (organisations or individuals) directly involved in the upstream and downstream flows of products, services, and/or information from a source to a customer"* (Mentzer *et al.*, 2001). Depending on the number of entities being part of it, the supply chain can have different degrees of complexity. The simplest supply chain would consist of a supplier, a company and a final costumer. Its degree of complexity could increase when adding more entities such as a supplier's supplier, third-party logistics suppliers (more commonly known as 3PL) or a customer's customer. In any case, the final costumer is considered part of the supply chain. It is important to consider that most of the organisations are involved in numerous supply chains and that its role is not necessarily the same in all of them. For example, one entity can be a supplier in one supply chain while being the customer in another one. (Mentzer *et al.*, 2001)

On the other hand, according to the *Business Dictionary*, logistics consists in the *"planning, execution, and control of the procurement, movement, and stationing of personnel, material, and other resources to achieve the objectives of a campaign, plan, project, or strategy. It may be defined as the 'management of inventory in motion and at rest'* (Bussiness Dictionary, 2019).

Sometimes the words logistics and supply chain management are erroneously used as synonyms. However, logistics is only one of the functions involved in supply chain management, as will be further detailed.

2.2. Supply chain management

According to the *International Center for Competitive Excellence*, "*Supply Chain Management is the integration of business processes from end user through original suppliers that provides products, services, and information that add value for customers"* (Cooper, Lambert and Pagh, 1997).

The definition of SCM may differ depending on the author. Nevertheless, virtually all of them agree that SCM considers the supply chain as a whole. All the organisations involved have the same aim: fulfil customers' requirements while minimising costs. The competence is not anymore between companies but between supply chains. It is also known that supply chains have always existed, but organisations being part of them used to work for their own business interests, without taking other entities into consideration. With SCM, this idea has been put aside. Nowadays, they all know that the activities being carried out individually have an effect on the other members of the SC. They all put effort into working in the most effective and efficient way in order to succeed. (NC State University, 2017)

Regarding the difference with logistics, and as mentioned before, logistics is only part of the supply chain management. In order to achieve customer's satisfaction, additional issues beyond logistics should be handled (Mentzer *et al.*, 2001). With this aim, SCM does not only include logistics, but also product development, marketing research, sourcing, production, promotion, sales as well as the coordination of all these activities using information systems (Cooper, Lambert and Pagh, 1997; NC State University, 2017).

2.2.1. Scope and objectives of SCM

The objectives of the SCM are, among others, the following ones:

– Reduction of the costs and the total amount of resources used (Mentzer *et al.*, 2001).
– Achieve customers' requirements (Mentzer *et al.*, 2001).
– Enhance customers satisfaction (Mentzer *et al.*, 2001).
– Synchronization between the requirements of the customer and the flow of materials from suppliers (Stevens, 1989).

- Improve efficiency and effectiveness (Mentzer *et al.*, 2001).
- Improve a firm's competitive advantage and profitability (La Londe, 1997).

2.2.2. Set of activities to implement SCM

After the description of SCM and its scope, it is easy to understand the concept of supply Chain Management as a process. According to *Defining Supply Chain Management of the Journal of Business Logistics*, seven activities should be carried out to succeed when implementing SCM (Mentzer *et al.*, 2001). This set of activities is shown in *Table 1* and commented below.

1	Integrated Behavior
2	Mutually Sharing Information
3	Mutually Sharing Risks and Rewards
4	Cooperation
5	The Same Goal and the Same Focus on Serving Customers
6	Integration of Processes
7	Partners to Build and Maintain Long-Term Relationships

Table 1. Set of activities to Implement a Management Philosophy (Mentzer *et al.*, 2001)

Thus, the organisations involved in supply chains should have an integrated behaviour. This means that they should not only think in their own profit but in the profit of the entire SC. Thinking of a supply chain as a single entity leads to include customers and suppliers in the process. This global mindset force organisations taking part in it to mutually share information. This helps to better organise each individual entity and, as a result, the whole SC. This gives rise to better rewards for all of them but also could result in significant risks.

As commented before, all entities have the same goal and cooperate with each other in order to satisfy customers and fulfil their requirements. Examples of cooperation would be sharing planning or having mutual activity control in order to evaluate the SC performance. Partners should also integrate their processes and form long-term strategic and tactical alliances.

However, companies could have difficulties when trying to successfully manage the supply chain where they are involved. For example, it is easy to find some conflicts of interest between companies. Not only day-to-day operational decisions should be made, but also strategic and tactical decisions, which involve medium-term and long-term decisions, respectively. Sometimes, and most especially with big supply chains, this can be difficult to handle and the communication between companies can become extremely complex.

It can also be difficult to deal with cultural differences when organisations are based in multiple countries. They may have, for example, a different way to treat the customer or to understand cooperation between companies. All the entities have to deal with that from the beginning and try to be open-minded and focus on what may be good for the whole supply chain and, as a result, to themselves.

2.2.3. Importance of data and analytics in SCM

Every company or entity collaborating in a supply chain generates a large amount of data, which is essential for the optimisation of the SC. But the most important thing is not having a large amount of data but knowing how to disregard useless data and to properly interpret the relevant data and making accurate decisions from it (Pontius, 2016).

With the analysis of data, some effects such as the bullwhip effect can be avoided. The bullwhip effect occurs due to the variance of orders in each stage of the supply chain. This variance starts in the lowest stage of the supply chain and keeps growing causing a large variation of demand in the highest stages (Logistics & Materials Handling Blog, 2012). It is also important to not spend too much time analysing data and forgetting other important issues that affect SC development.

As commented before, it is critical to know which data can be underestimated and which should be studied. According to *Aberdeen Group* research, 84% of companies have useless data whereas only 16% of companies have quality data. Besides, it is also known that 60% of a company’s time is spent in the identification, collection, and validation of data while only the 20% left is spent analyzing the model output itself (Koch, 2016).

In this way, it is important to know which kind of data is being collected and which can be the use when analyzing it. According to Koch, eight kinds of data can be collected from a supply chain:

- **Master data:** products and their relationships
- **Inventory data:** inventory volume and value and allocations between Distribution

Centers (DC's) in the network

- **Warehouse data:** storage capacity, storage characteristics, logistics equipment, personnel and contracts (outsourced).
- **Production data:** product portfolio and production capacity.
- **Volume data (or Logistics data):** suppliers to plant, plant to DC, stock transfer, plant-to-customer and DC-to-customer deliveries.
- **Financial data:** transportation, warehouse, production and inventory data
- **Demand data:** customer service, lead times and business requirements data
- **Qualitative data:** actual historical and forecast demand.

All these data can be somehow useful. Different kinds of data would be needed when trying to design a warehouse compared to when trying to design a supply chain (Koch, 2016). Companies should know which are their needs and therefore collect and analyze the most appropriate one.

Besides, according to *Accenture Research*, it is known that companies using analytics in their SCM achieve success in different areas, as shown in Figure 2. Although analytics in the supply chain is proven to be worthy, since most of the times it requires a considerable investment and generates security and privacy concerns, not many companies are willing to such a big change. Nowadays, only four out of ten companies have a true enterprise-wide supply chain strategy (Gii Finance Network, 2016).

Figure 2. Areas improved when using SC Analytics (Gii Finance Network, 2016)

In a nutshell, the smart use of data is a key factor when trying to optimise a supply chain. More important than having lots of information is knowing how to analyse and interpret it. Although it is proven that companies using analytics in their supply chains improve their performance, there is still a long way to go for companies to do it.

2.2.4. Uncertainty in the supply chain

Although companies are aware of the importance of SCM, it is inevitable for these to be affected by uncertainty. As previously commented, the scope of supply chain management is to reduce overall inefficiencies, and this includes trying to reduce the consequences generated by risksassociated with uncertainty, for example with variability of demand. If the demand is higher than the one forecasted, this could lead to a lack of stock and therefore to loss of sales. What is more, this uncertainty would easily propagate throughout the whole network of entities in the supply chain.

When planning how to manage uncertainty in the supply chain, companies need to deal with many issues. They should wonder, among others, how many units will customers order, how large should their stock be or whether they will receive the goods from suppliers on time. The decision-maker usually chooses to create safety buffers in time, capacity or inventory to be ready for variances. However, having these buffers reduces the competitive advantage and restricts operational performances (Patil, Shrotri and Dandekar, 2012).

In the article *Supply Chain Uncertainty: A Review and Theoretical Foundation for Future Research,* sources of uncertainty have been classified into three categories: internal organisation uncertainties, internal supply chain uncertainties and external uncertainties (Simangunsong, Hendry and Stevenson, 2012).

Internal organisation uncertainties are the ones which result from one company or organisation of the supply chain. This category includes uncertainties related to product characteristics, manufacturing process (machine breakdowns, process reliability...), inappropriate assumptions of a control system (for example in an ERP system), decision complexity due to multiple dimensions in the decision-making process, or organisational issues of the company.

On the other hand, internal supply chain uncertainties are due to the interaction between an organisation and the other partners of the supply chain. Some examples are irregular end-customer demand, demand amplification due to bullwhip effect, supplier issues (late deliveries, quality problems…), and order forecast errors due to long horizons. Uncertainties also increase because of the complexity of the chain configuration, which sometimes involves lots of parties, infrastructures and facilities.

Finally, external uncertainties are the ones which are outside a company's direct areas of control, such as political and environmental issues or natural disasters.

Although most of the times it is extremely difficult to predict them, some organisations try to find solutions to uncertainty. In the article *International Journal of Emerging Technology and Advanced Engineering*, some solutions to typical uncertainty cases are proposed. Returning to the example of the variability of demand, if a company knows that their demand is quite volatile, they have several options to manage it. For example, they can use flexible work hours to adapt the production with the demand, use temporary workforce during the peak season in order to increase their capacity or subcontract peak production. (Patil, Shrotri and Dandekar, 2012)

Besides, it is also recommended to centralise information in order to have demand details and inventory status updates instantaneously available, which might be very useful to manage distribution centres. Another solution to face uncertainties is to adopt a *postponement strategy*, which allows companies to delay some supply chain activities until customer demand. This could be implemented by having semi-finished products which could be easily customised in production facilities close to the customer or by keeping finished products in a central location that allows a fast distribution to the customers (Patil, Shrotri and Dandekar, 2012).

To sum up, it is of particular importance that companies are aware of possible uncertainties that they might be exposed to. They will be affected by them not only because of their performance but also because of taking part in a supply chain. It is important to have performance indicators that help to identify possible risks and uncertainties and to try to look for possible solutions that can reduce its effects.

3. Digital Twin

In order to introduce analytics to the supply chain, this thesis develops the concept of digital twin. A digital twin is a virtual replica of an object or process which simulates its real behaviour. The virtual twin contains all the properties and information of the real object or process in order to be identic to it. Through information obtained from sensors and automatisms, a virtual model of a product, process or even a service can be modelled. According to Marr, the connection between the physical and digital worlds allows data analysis and monitoring of systems to head off problems before they even occur, prevent downtime, develop new opportunities and even plan future scenarios by using simulations (Marr, 2017).

3.1. Object-based digital twin

An object-based digital twin is the first form of a digital twin that comes to someone's mind after reading the previous definition, and it is also the most widespread one. It consists of an object with smart components and sensors that gather data and send it to a cloud-based system. The data is analysed, and its interpretation is used to make future decisions. The digital twin allows us to generate a virtual environment which, based on the data received, new simulations can be carried out and future decisions can be taken in order to improve the real object performance (Marr, 2017).

Figure 3 shows a scheme that helps to understand better how a digital twin works. Data is first gathered with sensors and small components that are fitted to the real object. The real-time measurements are saved and transferred to a cloud, which is the link between the real product and the digital representation. The data collected is analysed with modern-day massive processing architectures and advanced algorithms. This allows the digital twin to build different environments and run several simulations varying the different parameters that define the model. After that, the simulations are evaluated and the results are saved and transferred to the cloud. The information from the cloud is adjusted to the real-world situation and it is send to the real product.

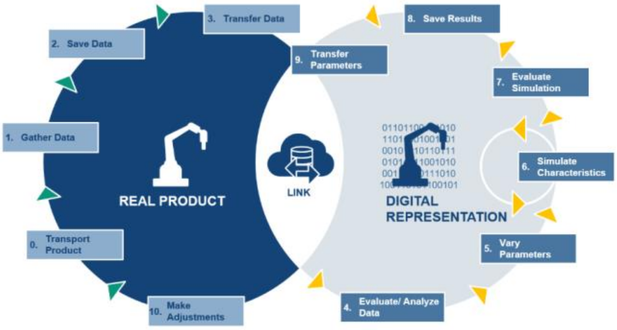


Figure 3. Digital Twin scheme (Unity consulting & Innovation, 2018)

Thanks to digital twins, it is easier to understand the products' performance and optimise it. What is more, a digital twin can improve customer satisfaction, troubleshoot and can help with product differentiation and product quality. Thus, it not only helps us to understand how products are performing but also how will they perform in the future. (Mikell and Clark, 2018).

As already commented in the previous chapter, data is only useful when well collected and organized, in a way that can help the decision-making process (Kitain, 2018). According to Kitain, the implementation of a digital twin in a product consists of the stages shown in Figure 4:

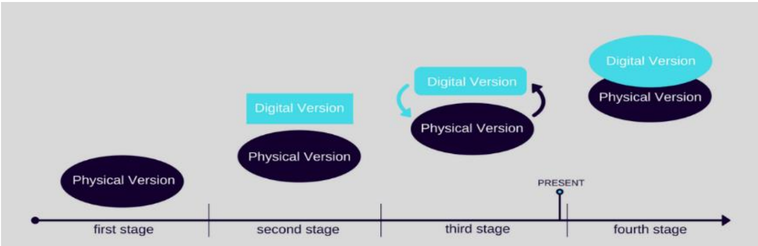


Figure 4. Stages of the Digital Twin evolution (Kitain, 2018).

In the first stage, only the physical version exists, and after that, a digital version is implemented (second stage). In the third stage, the interaction between both versions start but it is not until the fourth stage that the interaction goes further and both versions converge.

Kitain also claims that digital twins have an impact in three different zones: product and design, products in the field and future products. In the product and design zone, the digital twin can create different scenarios and simulate them by adjusting parameters without any risk in real production. It allows to predict failures and improve efficiency. In the second zone, where products are already in the field, digital twins allow lowering service costs and improving customer satisfaction. Configuration of products and problem diagnostic can be remotely done using the twin. Last, by analysing past behaviours in different scenarios, new products can be developed (Kitain, 2018).

Briefly, having an object-based digital twin allows us to virtually represent a product and by analysing historical data and creating several scenarios, optimise its whole lifecycle including, among others, efficiency, transparency, visibility, quality, and scalability. (Automation, 2019).

3.1.1. Examples of object-based digital twins

According to Gartner, 13% of companies implementing Internet of Things (IoT) projects are already using digital twins while 62% are either in the process of establishing them or plan to do so within a year (Gartner, 2019). This shows that, although not many companies are already using them, most of them are very interested and willing to invest in this technology. This section presentes three examples of companies that have successfully implemented object-based digital twins and that profiting from them. An example of a company that is currently implementing them is also presented.

Wärtsilä

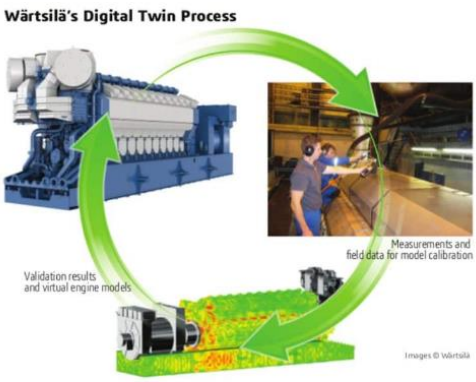


Figure 5. Wärtsilä's Digital Twin (ADA3DS, 2018)

Wärtsilä is a Finnish company that designs and manufactures four-stroke engines for cruises in 70 different countries. Their products are one of the biggest engines worldwide with a lifespan between 25 and 30 years. The company used to build prototypes for every motor in order to test and improve them, but its production was excessively expensive. Because of that, they decided to move to the digital world. Using models and 3D simulations they achieved better designs and avoided errors in the manufacturing process. In order to take the maximum advantage of this models, Wärtsilä went one step further and decided to implement digital twins. A scheme of its digital twin is shown in Figure 5. Nowadays, they install several sensors in the new engines that gather data regarding the engine's performance and feed the digital models and simulations. (ADA3DS, 2018)

Wärtsilä’s digital twin represents the operating conditions of the motors and by means of analytics, the company is able to work and improve in several areas: asset analysis and monitoring, predictive maintenance, predictive diagnosis, system performance optimization, machinery and system design, system testing and verification… (Wideskog, 2018)

Kaeser

Kaeser is a U.S. compressed air products manufacturer, and they also offer maintenance and operation services to the customer. Thanks to digital twins, they can offer not only preventive and corrective maintenance but also predictive maintenance. With predictive maintenance, the risk of performing the service too early or too late, as it happens with preventive and corrective maintenance, respectively, is avoided.

The digital twin allows the company to have real-time data to monitor the equipment. Thus, it is easy to detect potential faults in advance. Kaeser can maintain the asset throughout its lifecycle and charge fees based on air consumption. Sigma Smart Air, as the digital application is called, also includes monitoring of key figures such as service costs, reserves and specific power of the compressed air units. Moreover, according to the company, they have cut commodity costs by 30% and onboarded 50% of major vendors using digital twins. (Kaeser, 2018; Thomas Ohnemus, 2018). A scheme of how their digital twins work is shown in Figure 6.

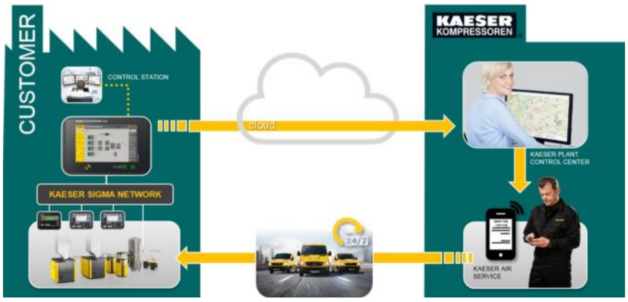


Figure 6. Kaeser's digital twin scheme (Kaeser, 2018)

Stara

Stara is a Brasilian company that provides innovative agricultural solutions to its customers. All their tractors are equipped with IoT sensors. Therefore, they can analyse the data gathered and simulate new environments, which allow them to increase equipment performance, prevent equipment malfunctions and improve asset uptime. Besides, Stara has launched a new service very useful for its customers: real-time insight detailing the optimal conditions for planting crops and improving farm yield. According to the company, farmers have reduced seed use by 21% and fertilizer use by 19% thanks to Stara's guidance (Thomas Ohnemus, 2018).

Onroak Automotive

Onroak Automotive is a French company, that designs, manufactures and sells racing cars. This project, which is currently in the second year of implementation and it is supposed to be completely implemented after the third year, has the aim of checking how cars are built and how drivers and mechanics are trained. The company believes that improving in those three dimensions can mean a great competitive advantage.

In this case, the digital twin not only helps in improving the design and manufacturing of cars but also it is intended to be used as a training for the pit stop. The digital twin allows the mechanics to exactly know where the spare parts are located and how are they supposed to be assembled. Thus, they can know, for example, where is enough space to put their hands while changing the wheels.

The company believes that with this information flow between the real object and the digital twin, they could save between 5% and 8% of combustible and even skip up to one pit stop (ADA3DS, 2018).

These are only four examples that try to show different fields where digital twins can be used and the improvements that companies experience when implementing them. Needless to say, digital twins can be implemented in other fields such as healthcare, robotics, customer experience or smart cities, among others.

3.2. Process-based digital twin

Although the idea of digital twin is often envisioned as an object, it can also be related to a process or service. As it is stated in Microsoft's article, "*the Process Digital Twin is the next level of digital transformation, compounding Product Digital Twin benefits throughout the factory and supply chain"* (Hanneman, 2017).

When a digital twin represents a single product, it can help to improve its lifecycle, but it cannot make decisions itself because other external factors are not taken into account. It will always be needed a human to consider what to do. In this article, Hanneman explains it with a good example. If we have an object-based digital twin of a machine and an error is detected, it may seem obvious that the next step should be to shut down the machine. But this decision should be taken by a qualified technician because it may cause severe consequences in the entire production line. The purpose of a process digital twin is to virtualize the whole process and the relationship between its components, allowing to optimize the entire production environment instead of only the equipment lifecycle (Hanneman, 2017). Hanneman's example makes explicit reference to a manufacturing process but it can also be extrapolated to a supply chain.

Figure 7 shows, also by means of an example of a manufacturing process, how a process digital twin works.

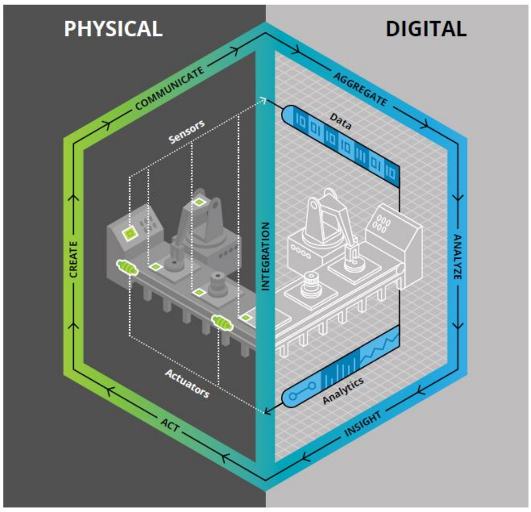


Figure 7. Manufacturing process digital twin model (Deloitte Unity Press, 2017)

It can be easily noticed that the process digital twin consists of the same components than an object-based one. This time, the sensors are distributed not only in one product but throughout the process. They capture both operational and environmental data from the whole process in the real world and send them to the cloud. Operational data involve all the criteria of the physical performance of the product, whereas environmental data refer to external data affecting the process operations. The data collected are usually combined with other data from the company, such as engineering drawings or the Bill of Materials (BOM). After data is integrated, it is processed and prepared to be analysed in the digital world, where insights are produced. Advanced analytics platforms and technologies are used to build models that will help the digital twin in the decision-making process.

The digital twin tries to find opportunities for saving costs, improving quality or increasing efficiency of the process, among others. At some point, after different scenarios are built and several simulations are run, it is considered that the digital twin is trained and ready to make its own decisions. It can decide which action should be done and by means of actuators instruct them to the real-world process. The training never stops because the process continuously sends real-time data to the digital twin, which keeps improving. Ideally, the digital twin should be able to properly make its own decisions, but they can be subject to human intervention (Deloitte Unity Press, 2017).

Returning to our field of study, process digital twins can be very useful for companies in a supply chain. They allow them to model different scenarios and run several simulations, which results are used for future decision making and predictions. The result is a change in companies' behaviour from taking reactive decisions to proactive ones. What is more, process digital twins can also be used to test new procedures. This allows making as many changes as wanted without the need of testing them in the real world, which usually leads to waste of time and resources (Andersen, 2019). The process digital twin removes the need to physically build the prototype, which is also an advantage with regard to object-based digital twins (Automation, 2019).

On the other hand, according to Hanneman, the fact of being able to model different scenarios enhances the benefits in the three following areas: machine level, factory level and supply chain level. Regarding our field of study, the author claims that it in the supply chain level is where process digital twins add the most value. They allow to quickly adapt to customer demands and accelerating the time to launch. They also have other benefits such as help manufacturers create a most robust horizontal value network, improve collaboration across suppliers and accelerating re-engineering processes from anywhere, without the need of being physically there. In the article, it is also claimed that companies that have already started implementing process digital twins are seeing benefits at every level. (Hanneman, 2017)

To sum up, when using digital twins, problems could be avoided before they occur, idle times could be predicted, new business opportunities could be developed, and production could be customized by the customer requirements.

3.2.1. Examples of process-based digital twins

As well as in the case of the object-based digital twins, it is possible to find some organisations that are already using process-based digital twins to improve their performance. The following section presents three examples of process-based digital twins.

Palembang GMA Refinery Consortium (PGRC)

PGRC is an international consortium of energy companies that develop, build and manage an integrated oil refinery. This project is on the process of implementation and its aim is to build a refinery in four years. The digital model will allow the company to optimise the design of the whole infrastructure and to organise different aspects of the project such as the construction process, coordination of suppliers, waste reduction, etc. This digital twin is not only intended to improve operational performance but also project management. According to the company, thanks to the digital twin, the investment needed for the construction of the refinery has decreased by 25% in comparison with the initial budget. Besides, it is estimated operational improvements of 15%-20% compared to the standards of the industry (ADA3DS, 2018).

AspenTech

AspenTech is a U.S. company that offers supply chain planning solutions. AspenTech uses digital twins to both offer maintenance solutions and to improve the whole factory scheduling based on an integrated digital twin maintenance model. According to the company, as one example, Aspen Mtell (AspenTech's digital twin) provides more than 25 days of advance warning of a central valve failure. This means that the planner has time for rescheduling and the costs will be lower than reacting to unplanned downtimes. The digital twin can trade off customer commitments, inventory holding costs and manufacturing costs. (Banker, 2018)

General Electric

General Electric (GE) is a U.S. multinational conglomerate that has decided to invest in digital twins to virtually monitor its supply chain. GE operates through aviation, healthcare, renewable energy and additive manufacturing, among others. More than 800.000 digital twins are already implemented in its plant in Minden (Nevada, U.S.A). According to Jeff Gordon, the plant manager, digital twins allow them to anticipate possible disruptions and to keep the company as productive as possible (Hernández, 2018). GE believes that virtually representing power plants is the best way to improve their performance. The digital twin keeps improving because the plant never stops operating. The digital twin also allows the company to make informed decisions regarding performance, assign loads and line-ups through time, and perform the right maintenance tasks at the ideal time. Besides, it allows "what-if" scenarios and therefore the company can make future predictions without wasting resources.

The digital twin designed by GE integrates analytic models for components and sensors from the plant with customer-defined Key Performance Indicators (KPI). Data are managed in a platform that allows the plant executives, plant managers and workers to interact with the digital twin in real time (General Electric, 2018).

3.3. Implementation of a digital twin

Implementing a digital twin in a company is somehow challenging. Most of the times it means having people fully dedicated to it that could be working instead on other short-term projects that, at first glance, could seem more important. However, when a company decides to invest time in building a digital twin, there are some steps that should be followed.

According to Kitain and Mussomeli, the process of implementation of a digital twin could be summarised in the following steps: envision, select, implement, industrialise, scale and analyse (Kitain, 2018; Mussomeli *et al.*, 2018).

First, the organisation needs to determine the optimal level of detail in creating the digital twin model. It is important to have as much information as possible but if there are too many sensors, this could lead to an overwhelming amount of data. Having too much data adds complexity and confusion to the decision-making process. Companies should find the equilibrium between a lack of information and be lost having to process a vast sea of data.

After that, the organisation should know why they need the digital twin, and what kind of benefits they expect from it. This can be easily done by imagining different scenarios in which the product or process that is going to be modelled could be exposed. The company should focus on these scenarios and try to think which outputs would like to have from the digital twin in each of them. Once this is done, a pilot should be chosen. The pilot should have a configuration that warrants success while also providing high output values. At this stage, it is not advised to go too deep with a complex digital twin, but to limit the scope of it.

Once the pilot is modelled, it should be implemented. After a while, the company will have the first insights from the pilot performance. At this phase, it is recommended to be open-minded to leverage new data collected during the process. Once the pilot has succeeded, it is ready to be shifted to an established tool. This could be done through improvements in the performance of the pilot.

Once the digital twin is successful, it is important to identify opportunities to scale in order to keep adding value to it. Lessons learned from the pilot can be used at this point. By means of advanced analytics, the digital twin can keep continuously improving in quality, efficiency, cost reduction and prevention of issues, among others.

3.4. Current situation and future expectations

By July 2017, digital twins were in the Innovation Trigger phase in Gartner's Hype Cycle. When a technology is at this stage, it generates interest in the market but neither the existence of usable products nor commercial viability is proven (Gartner, 2017). However, as it is shown in Figure 8, by August 2018 digital twins were already in the Peak of Inflated Expectations. This means that although many companies still have not started using it, some of them have successfully done it. This proves that the use of digital twins in companies is worth and that, in five to ten years, they will arrive at the Plateau of Productivity, where profits are generated.

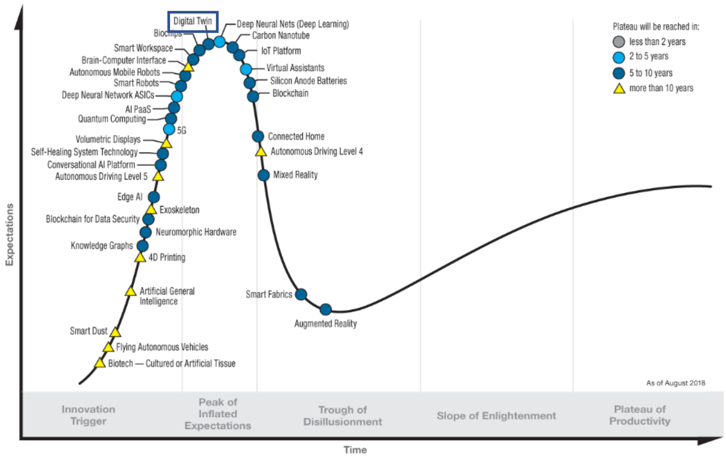


Figure 8. Hype Cycle for Emerging Technologies, 2018 (Gartner, 2018)

Thus, Digital Twins are growing faster than expected. According to a Gartner's survey carried out in 2018, by 2022 over two-thirds of companies that have already implemented IoT will have deployed at least one digital twin in production. However, with the current growth rate, this number might be reached within a year (Gartner, 2019).

On the other hand, and as already seen in chapter 3.2, when modelling a whole process several digital twins are involved. Companies do not only have to deal with the difficulties that appear when building a digital twin but also with the integration and communication between the different digital twins that take part in a process. If the communication between them is not properly implemented, the contribution of digital twins in the process could not be worth. Most of the times integration is complicated because it requires high-order integration and information management skills. However, companies are aware that integration is the key to success. According to Gartner, 61% of companies that have implemented digital twins have already integrated at least one pair of digital twins with each other (Gartner, 2019).

Regarding the companies that are still not using this technology, 74% of them say that are planning to do so in the next five years. Besides, according to Gartner predictions, by 2021 half of the major industrial companies will be using digital twins. This will result in an average efficiency increase of 10% (Gartner, 2019).

Thus, the adaptation of this technology in the market is being extremely fast. According to a market survey, the global market for digital twins is expected to grow 38% annually to reach $16 billion by 2023 (Markets and Markets, 2018). External factors also help in the development of this technology. First, IoT and ML are both proliferating very fast. According to Gartner, they are expected to be almost double by 2020 (Gartner, 2018). Second, several enterprise technology vendors such as IBM, Oracle, and SAP are offering digital twin solutions in the last two years (Mussomeli *et al.*, 2018). Both things help to accelerate the adaptation of digital twins in the worldwide market.

4. Methodology

Previous chapters have discussed the importance of supply chain management and the significant role that analytics play on it. In addition, it has also been discussed how digital twins can contribute to the optimisation of the supply chain. By using a digital twin simulation, this thesis is focused on dealing with one of the main problems that companies have when they try to improve their performance, which is inventory control. The aim of the project is to test whether a digital twin can beat a classical inventory control model.

First, the baseline model taken as a decision maker is described along with its assumptions and limitations. Subsequently, the concepts of neural network and reinforcement learning, which are needed to develop the training of the digital twin, are presented. After that, the experimental setup and the different factor combinations to test are embraced.

4.1. Baseline decision maker

The main problem of most companies is to have the correct amount of stock. If they have large amounts of items, the costs of holding them make companies erode their profits. However, if they have a lack of stock, they could lose transactions that could otherwise have occured. The solution is to find the optimal amount of inventory.

According to the *Merriam-Webster Dictionary*, inventory control is defined as the *"coordination and supervision of the supply, storage, distribution, and recording of materials to maintain quantities adequate for current needs without excessive oversupply or loss"* (Merriam-Webster, 2019). There are many inventory control methods which are currently used to deal with this issue. In order to choose the one which fits the most in a company, it is essential to know the demand behaviour.

Broadly, there are two kinds of demand models, deterministic and probabilistic. When a model is deterministic it means that all the given inputs will always result in the same outputs, without taking into account random variation. On the other hand, stochastic models are those who have at least one parameter affected by randomness and relationships between parameters are considered by means of probabilistic functions. (Barrera, 2016)

In this case, the Economic Order Quantity (EOQ) is taken as the baseline model because of its simplicity. Since the demand is known and constant, the EOQ model is considered a deterministic model.

4.1.1. Economic Order Quantity model

The Economic Order Quantity (EOQ) model was first developed by Ford W. Harris in 1913 and further analysed by R. H. Wilson. Because of that, it is also known as the Harris-Wilson model. The aim of this model is to find the optimal order quantity which minimises the cost of both holding and ordering items. This model not only answers the question of what quantity of items should be ordered but also when to order them (Sachin Agarwal, 2014).

The ordering costs are those involved when ordering additional inventories. In this model, it is assumed that the ordering cost per order remains constant. Thus, the fewer units requested per order, the more orders will be needed. Therefore, the total ordering costs will be higher. The holding costs are those incurred for holding inventory on hand. It is assumed to remain constant per unit of inventory. The more units in inventory, the higher the costs will be. The point would be to find the balance between both holding and ordering costs. The graphic representation of both costs and the total cost is shown in Figure 9. The optimal ordering point is determined by the intersection of the holding and ordering costs curves. At this point, they both have the same value and the total cost curve arrives at its minimum value (Kumar, 2016).

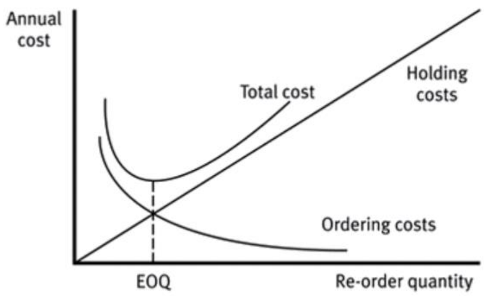


Figure 9. EOQ Costs (Kumar, 2016)

With the graphic above it is easy to understand how this model works. However, the mathematical formula that allows us to calculate the optimal order units is the following one:

$$Q^* = \sqrt{\frac{2 \cdot D \cdot S}{H}}$$

where D is the demand, S the ordering costs and H the holding costs. As a result, Q* is the optimal order quantity. The total costs and the order period are given by the formulas below, respectively:

$$Total\ Costs = \frac{D}{Q} \cdot S + \frac{Q}{2} \cdot H + D \cdot C$$

$$T = \frac{Q}{D}$$

where D is the demand, Q is the order quantity, S the ordering costs, H the holding costs and C the unit cost.

In Figure 10 it is represented the behaviour of the inventory level with time. At the beginning of the period (T) the inventory level is Q, and as time passes it decreases with the rate (D). The reorder point (r) indicates the moment in which the next order should be placed in order to have stock in time. The order should be placed taking into account the lead time (L) to exactly arrive at the same moment as the inventory level hits zero (Silver, Pyke and Peterson, 1998).

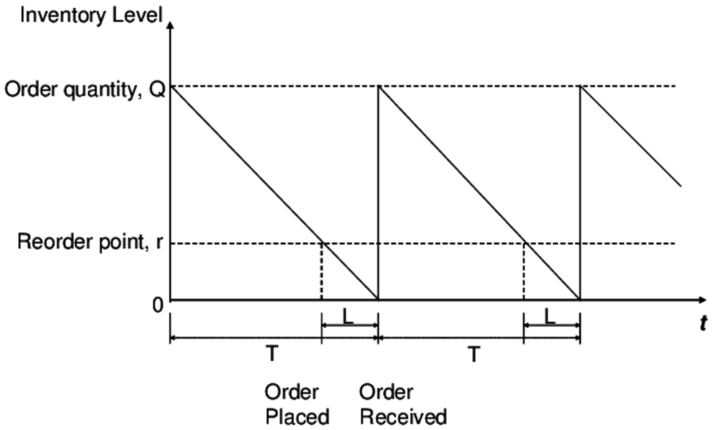


Figure 10. Behaviour of Inventory Level with Time in EOQ Model (Silver, Pyke and Peterson, 1998)

4.1.1.1. Assumptions and limitations of the model

The assumptions of the Economic Order Quantity model are the following ones (Sachin Agarwal, 2014):

- Demand is uniform, constant and continuous over the time.
- The lead time is constant.
- There is no limit on order size due either to stores capacity.
- The cost of placing an order is independent of size of order.

– The cost of holding a unit of stock does not depend on the quantity in stock.

In spite of the fact that the performance of this model is fairly satisfactory, there are several limitations since its assumptions are not an accurate description of reality. Demand and lead time are most of the times uncertain and exposed to changes. It also does not take into account that the ordering costs and holding costs may vary due to seasonal or economic fluctuations (Kumar, 2016). What is more, it is supposed immediately availability of the next order when the inventory level hits zero, which is a situation that could also differ from reality (Jose David Pinilla Manrique, 2011). Because of this, nowadays companies have a safety stock to avoid inventory shortage.

4.2. Digital Twin decision maker

In this thesis, a digital twin is trained to be in charge of the decision-making. The idea is that the digital twin would be able to make more accurate decisions than the classical model, which has some limitations. To accomplish this task, a neural network (NN) is used and re-trained with reinforcement learning until it is able to make its own decisions based on data and experience. In this section, the concepts of artificial neural network (ANN) and reinforcement learning (RL) are detailed.

4.2.1. Artificial Neural Network (ANN)

According to Dr. Robert Hecht-Nielsen, one of the first inventors of neurocomputers, a neural network is a "*computing system made up of a number of simple, highly interconnected processing elements, which process information by their dynamic state response to external inputs"* (Caudill, 1989). Artificial neural networks (ANNs), as their name suggests, are inspired by biologic neural networks, and their aim is to replicate the human learning process (Dormehl, 2019). In other words, they learn how to perform the same way humans do it: from experience. They learn by considering different examples, without being programmed with any task-specific rules (Ognjanovski, 2019).

ANNs consist of different nodes and layers. As it is shown in Figure 11, a node (or neuron) receives several inputs which have an associated weight (w_i). Its value depends on the relative importance to other inputs. The inputs can come from one or more nodes or from external sources. After that, the node takes the weighted sum of its inputs and applies an activation function in order to bring in non-linearity into the output. Since the real world data is not linear, the activation function is very important in order to have an accurate representation of reality. After data is passed through an activation function, an output is generated. This output will be the input of the next node. The signal always flows from the left to the right, until the last node is reached (Dertat, 2017; Ujjwalkarn, 2016).

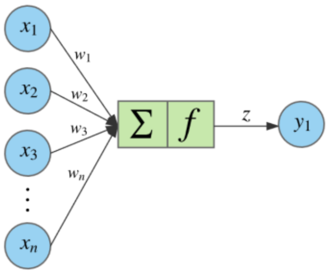


Figure 11. Single node or neuron (Dertat, 2017; Ujjwalkarn, 2016)

Nodes are organized into layers, as it is represented in Figure 12. A layer includes one or more nodes that operate together at a specific depth within a neural network. There are three kinds of layers: input layer, hidden layer(s) and the output layer (Ognjanovski, 2019):

- Input layer: it is the first layer and it contains the raw data, which comes from the external world. This layer sends the information to the hidden layers. No computation is done in this layer.

- Hidden layer(s): intermediate layer(s) where computation is done. As its name suggests, they do not have any connection with real world. This means that the output of this layer(s) is not visible since it goes to the next hidden layer or to the output layer. They learn about the data by minimizing an error/cost function.

- Output layer: last layer. It usually contains one node, which is the final output, but it can consist of several nodes. They send the information flow from the neural network to the external world.

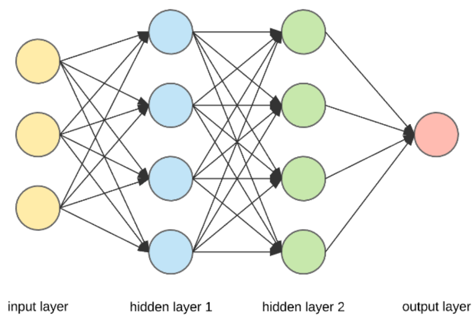


Figure 12. Neural Network Architecture (Ognjanovski, 2019)

As previously mentioned, the information flow goes from the nodes on the first layer to the node(s) on the last one. In the beginning, all the weights are randomly chosen. The network is trained with examples which result is known and it learns by predicting the results and re-adjusting the weights after each wrong prediction. When the NN is ready, it is used to make predictions which result is unknown (IBM Knowledge Center, 2014).

Artificial neural networks are widely used in machine learning applications to model complex patterns and prediction problems (Mahanta, 2017). In this Thesis, a Neural Network is used to train the digital twin, which intends to make its own order decisions.

4.2.2. Reinforcement learning

Machine learning (ML) is a field of artificial intelligence (AI) that tries to figure out how to perform important tasks by generalising examples. ML algorithms learn and improve from experience without being explicitly programmed (Domingos, 2016). There are mainly three categories of machine learning: supervised learning, unsupervised learning and reinforcement learning. The difference between the three methods is briefly defined below:

- **Supervised learning:** algorithms learn from labelled data. There is a whole host of examples from which computers can recognise patterns and associate them to new unlabelled data. After that, the algorithm is able to assign a label to the new data. Some examples could be: speech recognition, spam detection and handwriting recognition (Kent, 2018; Shetty, 2018)

- **Unsupervised learning:** algorithms learn from unlabelled data. This model has the ability to predict only based on a set of data. Since no categories are provided, it clusters information according to their similarities and differences. Some examples are: detect morphology in sentences and classify information. (Kent, 2018; Rouse, 2016)

- **Reinforcement learning:** algorithms try to optimise the solution of a problem by means of trial and error (Gomez, 2019). Some examples are: autonomous vehicles, chess game and decision making (Kent, 2018). This concept is widely explained below.

In this thesis, the digital twin needs to be able to make its own decisions in order to have the optimal amount of inventory. Because of that, reinforcement learning (RL) is the method chosen to train the digital twin.

The scheme in Figure 13 offers a representation of how RL works. Its main elements are an agent and the environment. The general idea is that an agent performs actions that affect the environment and receives rewards as a consequence, which can be positive or negative. Through observations and given the current state of the environment, the agent can figure out how the environment is. Thus, the agent learns which actions lead to positive or negative rewards (McMahon, 2018).

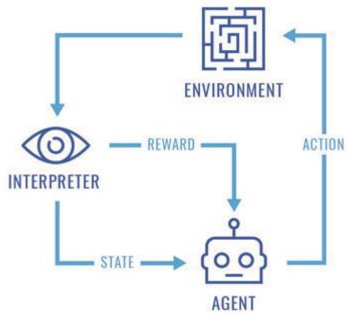


Figure 13. Reinforcement Learning Scheme (Gravelle, 2018)

The loop output results in a sequence of state, action, reward. The agent perceives a state s_t and performs an action a_t. Consequently, the environment responds with a reward $r_t = r(s_t, a_t)$. The result is the successor state $s_{t+1} = \partial(s_t, a_t)$. Both the reward and the successor state only depend on the current state and action. The agent's task is to learn a strategy or policy (π) for choosing actions. The policy defines how the system behaves in each time step ($\pi(S) \rightarrow A$) (Morales, 2016; Schmid, 2005). The aim is to maximize the cumulative reward R_t:

$$R_t = \sum_{t=0}^{\infty} \gamma^k r_t \qquad where\ \ 0 \leq \gamma \leq 1$$

This function considers another parameter: the discount rate γ. It determines the relative importance of future versus immediate rewards (Schmid, 2005). The aim is to discount each reward by gamma to the exponent of the time step. This means that when the discount factor is low, the agent will focus on short terms rewards because the discount will be bigger and, consequently, the future rewards will have less importance. However, if the gamma is large, the discount will be lower and the agent will look for long term rewards (Larson, 2018; Schmid, 2005).

Briefly, the agent's aim is to reach its optimal performance. In order to do it, it trains in the environment and modifies its behaviour by considering the accumulated rewards, which allow it to get to know the environment. The training finishes once the agent is capable of making the optimal sequence of actions that lead to the greatest accumulated reward (Gomez, 2019).

On the other hand, it is also important to highlight that there are two kinds of tasks in reinforcement learning: episodic tasks and continuous tasks. Episodic tasks are those which have a terminal state. This means that once a particular state is reached, the interaction between agent and environment finishes. The interactions agent-environment are called episodes and they consist of a list of states, actions, rewards and new states. On the contrary, continuous tasks are those that do not have a terminal state, the agent keeps running until it is asked to stop. (Dahiya, 2019; Larson, 2018)

Using reinforcement learning in the supply chain can help to improve its performance. But it is somehow challenging since it is not always easy to gather enough data to adequately understand the complexity of the supply chain environment. Besides, reinforcement learning can lead to overfitting the model to the training data and performing poorly in reality. Some companies have already started using reinforcement learning to improve its supply chain. An example of it is Streamba, a company that enables leading businesses to put their data to work (Gravelle, 2018; McMahon, 2018).

4.3. Experimental Setup

As previously mentioned, the performance of a digital twin simulation is tested and compared to the results given by a baseline model, the EOQ model. In order to do so, a code representing a digital twin simulation has been provided by Mr Tino T. Herden, this thesis' supervisor. The code has been adapted to include the reliability parameters in the neural network, thus allowing the performance of all the experiments. The adapted code can be found in in the section Code: Digital twin simulation of the Annex, and a high level explanation of it is given below. After that, the factor combinations that are tested are presented. The aim of the experimentation is to test whether the digital twin can beat the baseline model in different situations.

The code is divided into different sections: ordering functions, simulation, helping functions, training and experimentation. Each section along with its functions are explained below. Besides, the general performance of the code is described by means of a flowchart in order to understand the experimentation procedure.

Ordering functions

In this section, three ordering functions are presented: EOQ function, Random orders and Neural Network decision engine. The EOQ function needs the inputs described in Table 2 to calculate the optimal order quantity (Q*) using the formula detailed in chapter 4.1.1. Once Q* is known and considering the periods assumed to be per year, this function also calculates when should the units be ordered. The output is a vector which indicates the quantity to order and in which period should be ordered. For example, if Q* is ten units and they should be ordered in the first and sixth period of one year simulation that it is assumed to have twelve periods, this function returns a vector of twelve components, which all of them are zeros except for the first and sixth component, which are ten.

Inputs	
Demand	Annual demand (units per year)
ordering_cost	Cost per purchase order
interest	Holding cost rate (%) per unit per year
Cost_purchase	Cost per purchase unit
periods_in_year	Periods assumed per year
return_periods	Periods to be returned

Table 2. EOQ function inputs

On the other hand, the Random Orders function creates random orders using a uniform distribution. This function is used to create the initial training data for each trial and its inputs are detailed in Table 3.

Inputs	
Demand	Annual demand (units per year)
Demand_sd	Demand Standard Deviation
periods_in_year	Periods assumed per year

Table 3. Random Orders function inputs

The output is 50% of the times 0, which means that nothing is ordered. The other 50% of the times the output is a number which indicates the units ordered. In this case, a normal distribution is used to generate the orders considering the demand and its standard deviation. It is also taken into account that, since only half of the times the orders are created, the units ordered should be twice the demand of that period.

Finally, the Neural Network decision engine function, which is based on the trained neural network, and tries to find the order quantity with the lowest costs. The inputs of this function are shown in Table 4. Considering historic costs, historic demand and historic reliability, predictions are made for several order sizes and the one with the lowest costs is chosen as the output of the function.

Inputs	
NN	Neural Network
Demand	Annual demand (units per year)
h_cost_mu	Mean historic costs
h_demand_mu	Mean historic demand
h_demand_sd	History demand standard deviation
inventory	Inventory
h_delivery_accuracy_mu	Mean historic accuracy reliability (only in experiment 2)
h_delivery_ontime_mu	Mean historic delivery on time reliability (only in experiment 3)

Table 4. Neural Network decision engine function inputs

Simulation function

As its name states, this function simulates the sales, costs, inventory, etc. for a certain number of periods based on a chosen ordering function (EOQ, Random Orders or NN). The inputs of this function are shown in Table 5.

Inputs		Starting value
sim_type	Ordering function to simulate (EOQ, Random Orders or NN)	EOQ
Demand	Annual demand (units per year)	12000
Demand_sd	Demand Standard Deviation	600
periods_in_year	Periods assumed per year	100
return_periods	Periods to be returned	100
ordering_cost	Cost per purchase order	10
interest	Holding cost rate (%) per unit per year	0,1
Cost_purchase	Cost per purchase unit	5
Cost_lostsales	Cost per lost sale	7
var_rel	Varying reliability. Percentage of delivered goods	1
delivery_ontime	Probability of delivering the goods on the correct day vs one day late	1
h_length	Number of historic observations created	100
NN	Neural Network	NULL

Table 5. Simulation function inputs

Apart from the inputs that are needed to run the different simulation types, it is worth noting that there are two inputs to modify the delivery reliability. Var_rel refers to the percentage of delivered goods. If this parameter is one, 100% of the goods are delivered. On the other hand, delivery_ontime is a variable based on the probability of receiving the order on time versus one day late. To make it simple, it is a binary event. This means that there is no intermediate stage, either the goods are delivered on the correct day or they are delivered the next day.

The values shown in the third column of Table 5 are the ones chosen as the basis for the experimentation of this thesis. The starting values are the ones proposed with the code given and have been chosen in a way that tries to create a possible scenario in the real world. The first trial has been conducted with these values. In the following trials, some of these parameters will be modified in order to see whether the digital twin is able to make good decisions in different scenarios.

First of all, the simulation set-up is prepared based on the simulation type chosen: EOQ, Random orders or NN orders, and the daily demand for the return periods desired is created based on a normal distribution. A normal distribution is used in order to consider demand variability, which it approaches more to the real world. After that, historic data are created: demand, delivery on time, delivery accuracy and costs. Once the history is set up, the simulation is run for the number of periods in *return_periods*. Every simulation includes the following steps:

1. **Update inventory.** Inventory updates are done in the following order: late arrivals from the previous day are considered, new orders are generated, delivery reliability is applied and inventory is updated with eventual orders.

2. **Experience demand.** Sales are recorded. Sales will be the minimum between the stock in inventory and the demand. It is obvious that if the demand is higher than the stock, it is impossible to satisfy the demand and only the units that are in stock can be sold. After that, inventory is updated again by decreasing the sold units and, in case the demand is not met, lost sales are recorded.

3. **Calculate costs.** Costs are calculated, including costs of supply, ordering, lost sales and inventory.

4. **Update history.** Demand, reliability and costs of the current period are updated in history data.

5. **Recording all data in a list format.** At the end of the simulation, all data are recorded.

The output of the function is a data frame recording all data generated during the simulation. The data frame includes: daily demand, sales, lost sales, inventory level, orders, incoming goods of the period, varying reliability, number of items that arrive one day late (from previous period) and costs (supply, ordering, lost sales and inventory). It also includes mean historic data of demand, demand standard deviation, delivery accuracy, delivery on time and costs.

Helping functions

Three helping functions are used to ensure a good performance of the training of the digital twin: Normalization, Additional Normalization and De-Normalization. These functions are used to normalise and de-normalise data. The normalisation used is min-max normalisation, which allows having the entire range of values of X mapped to the range from 0 to 1.

$$Norm(x) = \frac{x - \min(x)}{\max(x) - \min(x)}$$

The de-normalisation function allows returning data to their normal values.

Training

The training is divided into three parts: the creation of the initial training of data, training of the neural network and reinforcement learning. First of all, the initial training data for each trial needs to be created. To create this set of data, 20 simulations using the Random Orders function as simulation type are run. After that, the neural network is trained for the first time. In order to create the neural network, data are first normalised. After that, the *neuralnet* package is used to create the NN. The NN has three hidden layers with five, three and one neurons, respectively. The activation function used is logistics (or Sigmoid activation function), which has a range from 0 to 1 and it has an S-shaped curve. This function has the form shown in Figure 14 and it is defined by the following formula:

$$f(x) = \frac{1}{1 + e^{-x}}$$

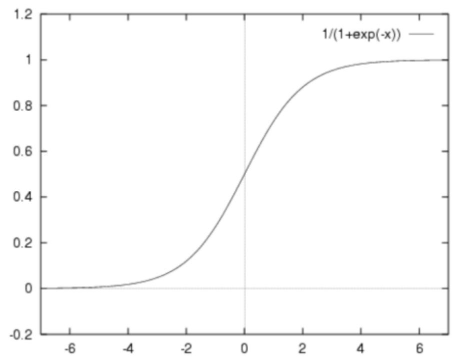


Figure 14. Sigmoid activation function

The model and the normalised data are saved. These two functions are only run every time that a new experiment is raised. As mentioned in the previous chapters, the model is retrained with reinforcement learning. Several iterations are run, and the results are compared with the ones obtained with the classical model. The training is supposed to finish when it is considered that the model is able to make its own decisions.

Experimentation

In order to analyse the model, 50 runs of experimentation are set for both EOQ and NN experiments. Two graphs with the results obtained are plotted:

- Mean inventory level during the entire course of the periods.
- Histograms showing costs: counting bars that display the distribution of costs during the 50 experiments.

General performance of the code

The general performance of the code is shown in Figure 15 by means of a flowchart. In each trial, new factor combinations are chosen and assigned to the parameters in the Simulation function. After that, an initial set of training data is created, and the NN is trained for the first time. These two steps are only needed for completely new trials. After that, the model can be retrained as many times as necessary. The number of iterations in the reinforcement learning process must be chosen. When the iterations are done, 50 experiments for the same NN and for the EOQ model are run. The aim of the thesis is to analyse the learning process of the model and test whether the digital twin can be a better decision maker than a classical model. When the results of the NN experiments are better than the EOQ experiments, the training is supposed to be over.

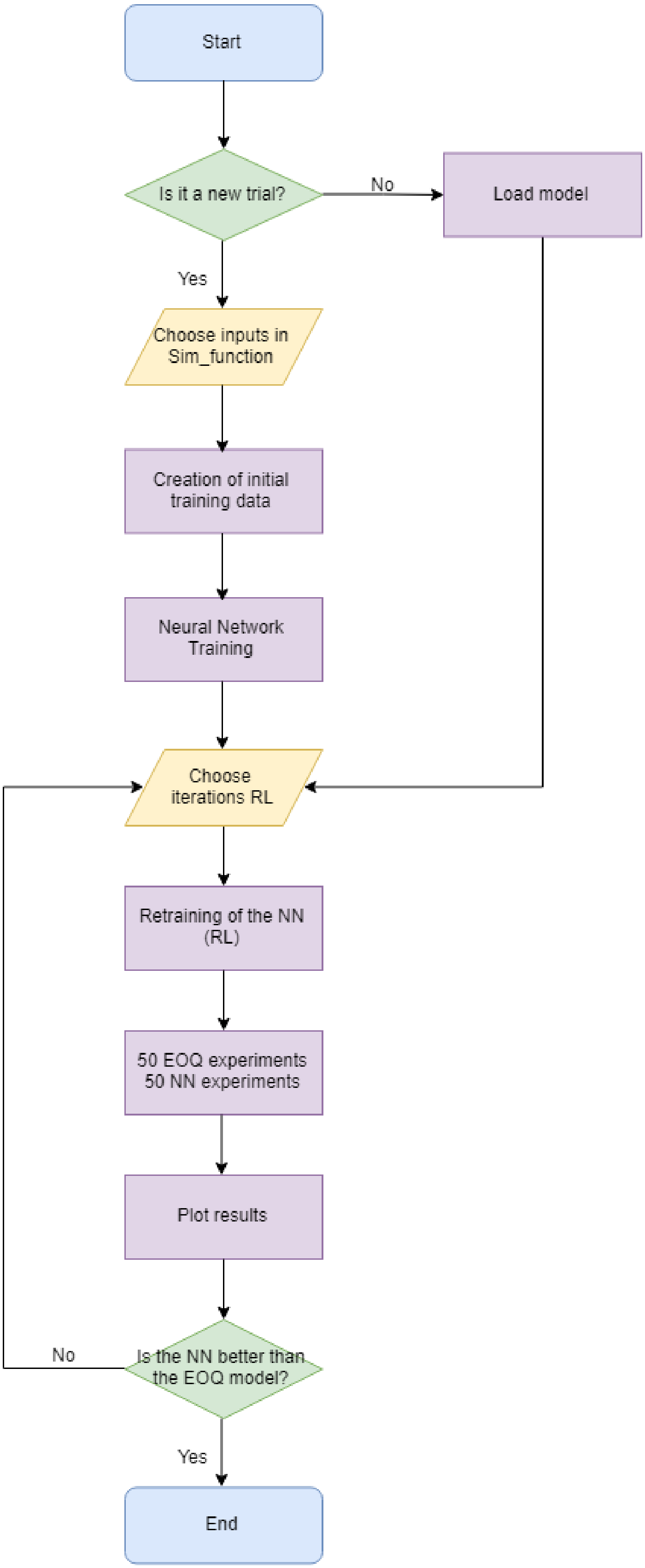


Figure 15. Experimentation procedure

4.3.1. Factor combinations to test

Once the methodology of the experimentation has been detailed, the different factor combinations to test are presented. As previously mentioned, the aim of the project is to analyse the learning process of the digital twin and test whether it is a better decision maker than a classical inventory model. In this thesis, three trials are carried out.

As detailed in Table 5, the simulation function has several inputs. The trials are the result of combinations of them. In order to choose which factor combinations should be used, it has been considered that the main point is to challenge the EOQ model by disobeying its limitations. In order to do so, the parameters chosen to be modified are *var_rel* and *delivery_ontime*.

The EOQ model, as previously detailed in chapter 4.1.1.1, considers that once the inventory level hits zero, the next order is immediately available. This could differ from what may happen in reality because lead time and demand are affected by uncertainty. Besides, even if the next order is available at the same moment that the last item of inventory is sold, it is also uncertain whether 100% of the goods will be delivered. And this is what *var_rel* and *delivery_ontime* try to represent. They are both thought to be delivery reliability indicators. On the one hand, *var_rel* refers to the percentage of delivered goods. On the other hand, *delivery_ontime* refers to the probability of receiving the goods on the correct day versus one day late.

The values chosen for these two parameters in each of the trials are the ones shown in Table 6. All the other inputs of the simulation function remain constant during all the experimentation process. In the first trial, both parameters are set at 1. This means that all the goods are delivered and that they all arrive on time. This would be an ideal situation.

After that, in the second trial, *var_rel* is modified but *delivery_ontime* remains constant. The value of *var_rel* has been decided to place a limit in a logical way. This means receiving the totality of the items most of the times but failing sometimes. In order to do so, a vector representing the reliability of each period is generated. The accuracy reliability is variable and it is a value between 0,85 and 1. This means that the percentage of delivered goods in each period will be between 85% and 100%.

In the third trial, *var_rel* is set at its default value and *delivery_ontime* is modified. The value of delivery_ontime represents the probability of receiving the goods on time. In this experiment, 100% of the goods will be received but only around 85% of the times the order will be received on time. As previously mentioned, it is a binary event. To represent it, random numbers between 0 and 1 are generated and compared to the limit by *delivery_ontime* parameter. If this number is higher *delivery_ontime*, a delay will be considered.

	Trial 1	Trial 2	Trial 3
var_rel	1	0,85 - 1	1
delivery_ontime	1	1	0,85

Table 6. Factor combinations to test

This experimentation seeks to highlight how both parameters individually affect the decision-making process. In each trial two things are studied: the learning process of the digital twin and whether it can overcome the classical model or not.

5. Results

This chapter presents the results of this thesis, and it is divided in two parts. First, the expected results before the experimentation are discussed and, after that, the results and conclusions for each of the trials are exposed.

5.1. Expected results

The model chosen as a baseline is the Economic Order Quantity model. This model has some limitations that reveal that it is not an accurate representation of the real world. The demand is assumed to be constant and continuous over time and the lead time is also considered constant. However, they are both uncertain in the real world. Besides, other factors that can affect the ordering process are not considered. As already discussed, a clear example of it is the delivery reliability.

The purpose of creating a digital twin simulation is being able to make better decisions by considering other factors that classical models do not take into account. The digital twin simulation learns with reinforcement learning. This means that it makes future decisions based on experience and that it is not tied up to any rule. For this reason, it is expected that the digital twin, after being trained, will become a better decision-maker than a classical model, which is unable to understand the environment and will be always restricted to its limitations.

As it only learns from experience, the digital twin can get to know the environment and considers all the factors that have historically affected inventory control. Contrary to that, the EOQ model calculates the optimal order quantity only based on a formula, without taking into consideration the supply chain's uncertainties. For example, two companies with the same demand and costs but based in different locations might have different suppliers. This means that they will be both affected by uncertainty but in a different way. In this case, the EOQ model does not notice any difference between the two companies but the digital twin does. This means that probably the digital twin makes different and more accurate decisions for both cases whereas the EOQ decision-maker cannot.

Figure 16 and Figure 17 provide a qualitative representation of the expected learning process of the digital twin in comparison with the one expected from the EOQ model. Figure 16 shows the cost distribution. In the beginning, it is expected that the digital twin makes erroneous decisions that imply high costs. After training, the curve is expected to move left approaching the EOQ model's curve. At the point where both curves overlap, highlighted in orange, the digital twin starts to behave like the EOQ model and can, in some cases, find the order quantity that leads to *optimal* costs according to the EOQ model. However, it is expected that the digital twin curve will be able to move even more to the left and offer a solution involving lower costs than the baseline model.

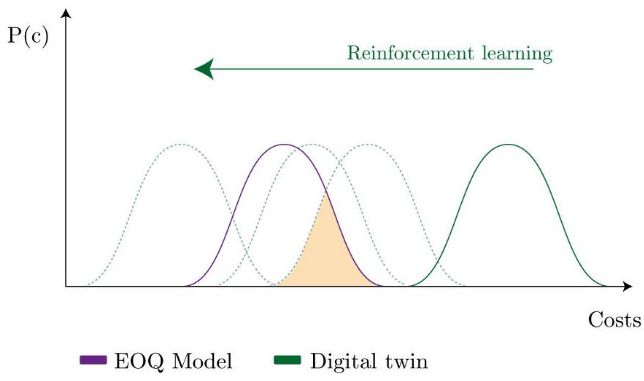


Figure 16. Distribution of costs. EOQ model VS Digital Twin simulation

Along with the costs, as shown in Figure 17, the inventory level is also expected to lower after the learning process, approaching the EOQ curve and become even lower. Before the simulation starts, the inventory level is initialised. The value given to it is the result of the EOQ formula, regardless of the simulation type. Because of this, both the EOQ model and the digital twin start at the same point in the graph. It is expected that at the beginning, when the digital twin is not able to make good decisions, the inventory level increases because more items than the ones needed are ordered. After training, the inventory curve is supposed to stabilise at lower values, approaching the EOQ model's curve.

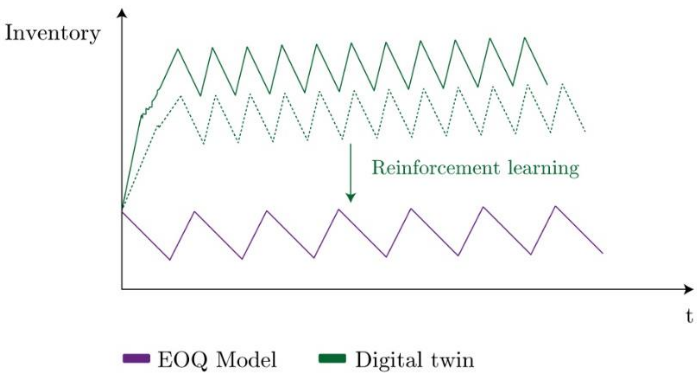


Figure 17. Expected inventory level. EOQ Model vs Digital Twin simulation

Although the general behaviour forecasted is the one described above, the results are expected to be different depending on the experiment. When the EOQ model is affected by uncertainty (trials 2 and 3), which is not contemplated by the classical model, it is assumed that the distribution of costs will be higher. Therefore, the EOQ model in these experiments is expected to be more easily beaten by the digital twin. Besides, the fact of receiving fewer items than ordered or receiving them late will be also reflected in the inventory graphs.

5.2. Experimental results

The following sections analyse the results of each trial. The methodology followed is the same for all of them. Since the training requires prepared data, every time that a new trial is performed, a first set of random data is created with 20 simulation runs. After that, the neural network is created. At that point, the digital twin is ready to be re-trained. The first time, 30 iterations of training are run, and the model is saved. From that moment, every 25 iterations, the model is saved and 50 experiments for the same neural network are run. The results of these experiments are plotted, and two different graphs are obtained: the mean of the inventory level and the distribution of costs. These two plots are compared with the EOQ model results. When it is considered that the neural network makes accurate decisions, the training is over.

Due to a matter of time and to the scale of this project, the models are trained with 325 iterations. All the graphs obtained from the learning process, as well as the graphs and calculations used to analyse the data resulting from experiments, are attached in the section Analysis of the results of the Annex.

5.2.1. Trial 1: var_rel = 1 / delivery_ontime = 1

In the first trial, var_rel and delivery_ontime are both set at 1. This means that 100% of the goods ordered arrive and that they do it on the correct day. The results obtained after 50 EOQ experiments are shown in Figure 18. The experiments result in a cost of 60.000 on average. It can be easily noticed a *triangle-shaped* inventory graph, as it was theoretically described in chapter 4.1.1. Nevertheless, since the graph is the mean of 50 experiments and the demand has been described by a normal distribution, its shape has been slightly affected. The inventory level remains more or less stable between 100 and 800 during the simulation period. Since in most of the experiments it rarely hits zero, the items accumulate, and the result is a slight upward trend.

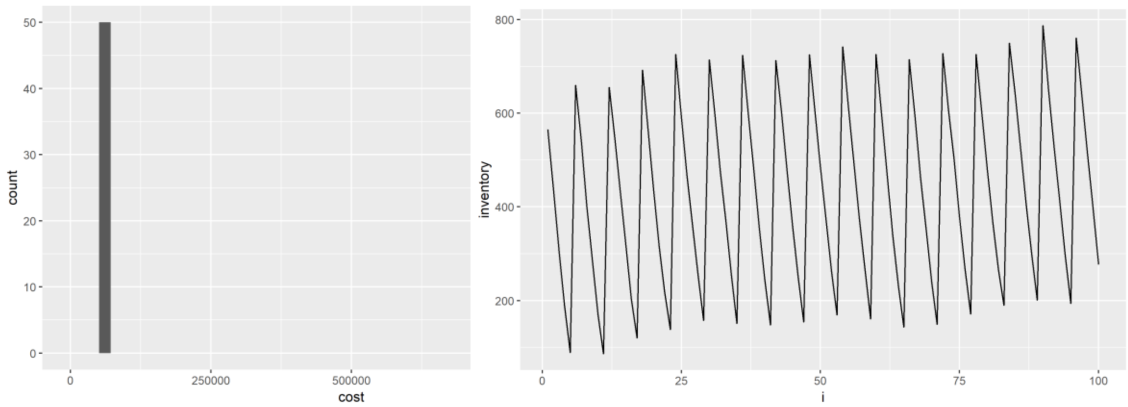


Figure 18. EOQ results for trial 1

The EOQ model does not consider demand variability. Because of this, orders placed in all the experiments are the same and, consequently, supply and ordering costs too. However, what makes the difference between the experiments are the inventory costs and the lost sales costs. Table 7 shows the results for the experiments with the lowest and the highest costs. It can be easily noticed that the costs due to the loss of sales is what differentiates them. Whereas in simulation run 6 they are only 105, they go up to 10.672 in simulation 12, which means a 16,1% of its total costs.

Simulation run	Supply costs	Ordering costs	Lost sales costs	Inventory costs	Total costs
6	55.440	160	105	270,59	55.975,59
12	55.440	160	10.672,5	166,22	66.438,72

Table 7. Comparison between simulation run 6 and 12

Regarding the digital twin behaviour, the learning process in this first trial is somewhat different to the one expected and detailed in chapter 5.1. It is important to note that, as previously commented, the simulation is forced to start with an inventory level calculated with the EOQ formula. In such a way, all the experiments start with the same inventory level.

As it can be seen in Figure 19, after only 50 iterations, the distribution of costs is not very far from the one obtained with the EOQ experiments (see Figure 18), but the costs are still higher. The number of items ordered in the first periods is significant. After a few periods, the inventory level decreases, and it becomes stable, but it is still high due to the first huge orders.

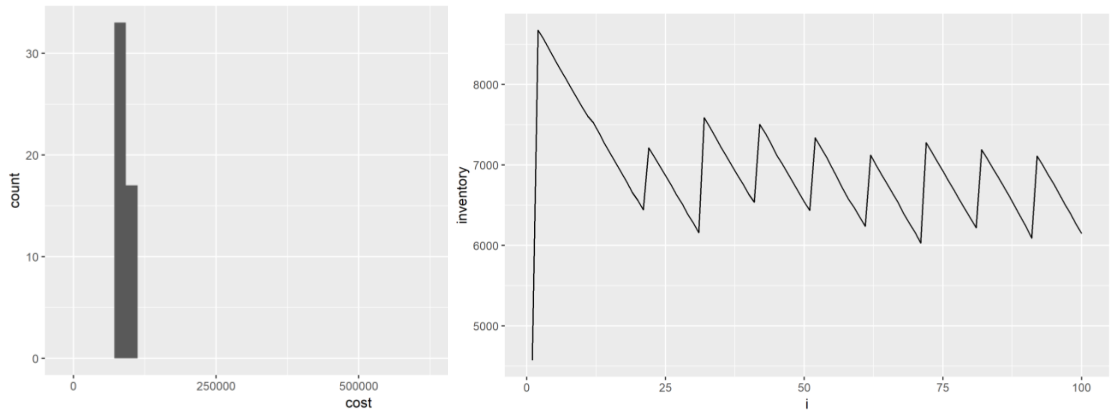


Figure 19. NN results for trial 1 (50 iterations)

Seeing these results, it seems as if the digital twin is learning faster than expected, but after some iterations more, it is obvious that the decisions made after 50 iterations are not stable and, therefore, not reliable. After 25 iterations more (75 iterations), the inventory level is even higher and the costs too. Contrary to expectations, instead of starting to slowly decrease in inventory level and costs, there is a sudden change. In iterations 100 and 150 the costs are low again, which could seem, a priori, a good trend. However,this is caused because the digital twin decides not to order at all (see Figure 20).

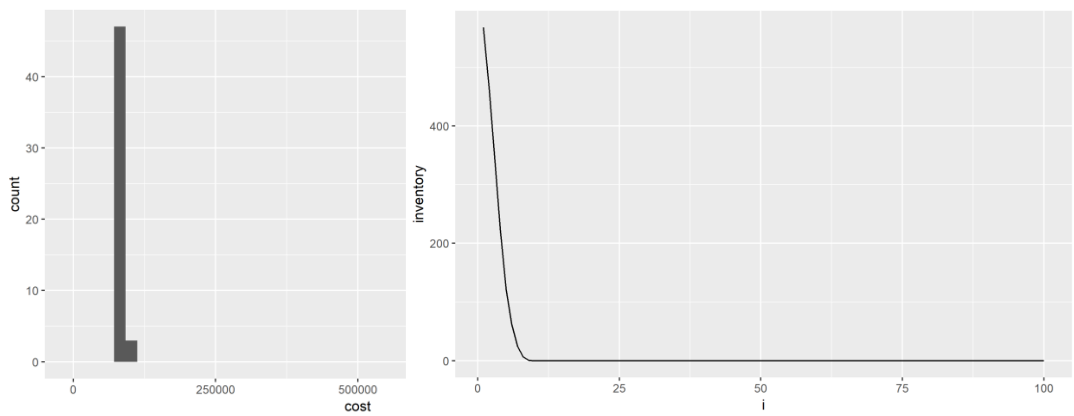


Figure 20. NN results for trial 1 (100 iterations)

Stock is available during the first periods because the simulation starts with some inventory, but no more orders are placed. This means that once the inventory level hits zero, it remains in zero. Obviously, this is not a feasible solution of a company.

After 25 iterations more, the experiments result again in high inventory level and high costs. From that moment, the digital twin starts to behave as expected. Both the inventory level and the costs decrease after several trainings. However, from time to time, the digital twin decides again to react as in Figure 20.

After 275 iterations, as shown in Figure 21, a good approach to the EOQ model is finally achieved. Regarding the cost distribution, 94% of the experiments result in the same costs as the EOQ model, whereas the 6% left are a bit higher.

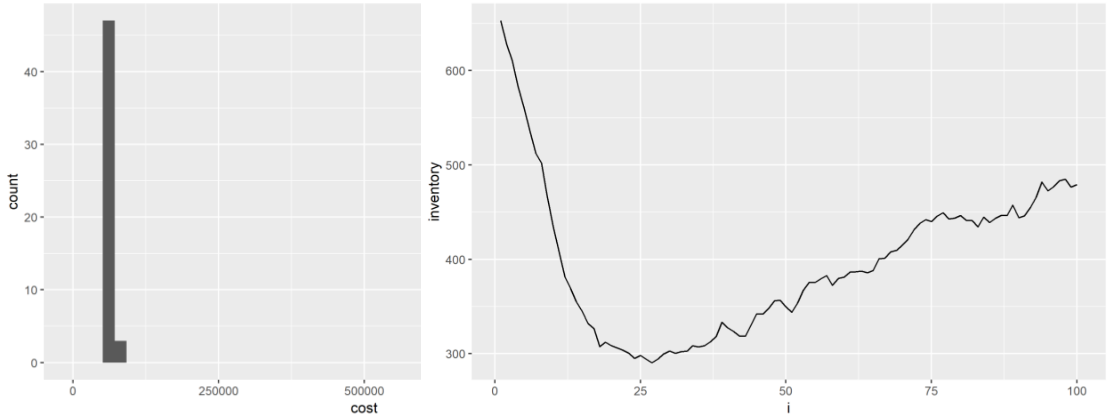


Figure 21. NN results for trial 1 (275 iterations)

On the other hand, the inventory level is significantly lower in comparison with the previous experiments. Although its values are reasonable, it is a bit unstable. After a big decrease, from period 25 on, an upward trend is visible. Because of the irregular shape of the graph, it is noticeable that the number of items ordered in each period is very variable.

Fifty iterations more are run before the trial is completed. In iteration 300, the digital twin adopts again the same behaviour as in iteration 100 (see Figure 20). However, as it is shown in Figure 22, in iteration 325 the digital twin's behaviour is again somehow similar to the one adopted after 275 iterations (see Figure 21). The trend in both cases is a decline of inventory at the beginning followed by a moderate accumulation of it.

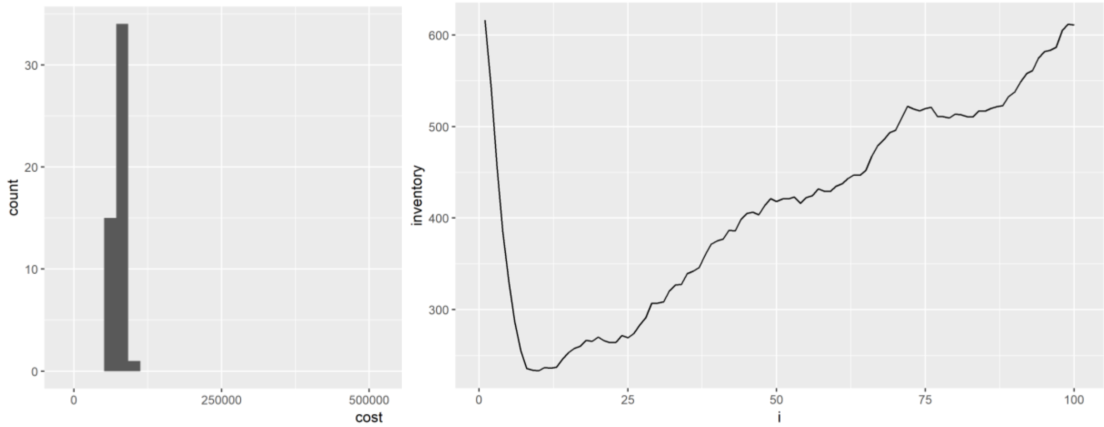


Figure 22. NN results for trial 1 (325 iterations)

Figure 22 shows that 15 out of 50 experiments lead to approximately the same costs than the EOQ model. These experiments along with the 5 experiments that lead to the highest costs have been deeply analysed and compared.

The behaviour adopted by most of the experiments that lead to high costs is not ordering at all or ordering from time to time. In these cases, an average of 92% of the costs are due to the loss of sales. An example of it is shown in Figure 23, where the inventory level for the experiment with the highest costs is represented. Although not ordering is the most common reason of high costs, they can also be due to ordering too much and not being able to sell it, as it is shown in Figure 24.

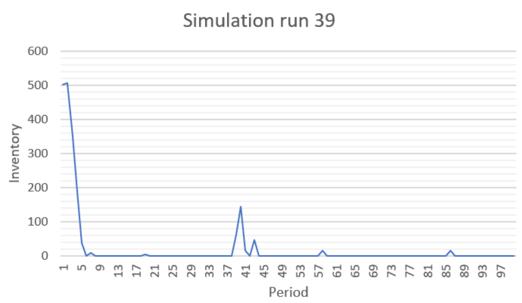


Figure 23.Inventory level for simulation run 39

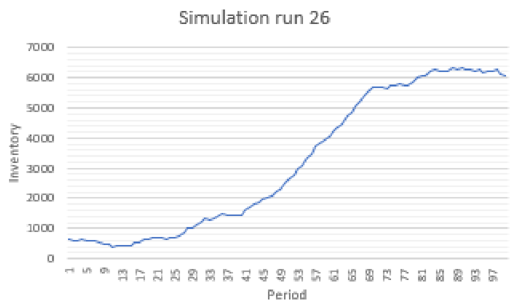


Figure 24. Inventory level for simulation run 26

Separately analysing only the experiments that lead to the lowest distribution of costs, it is easily noticed that the inventory level graph is slightly different as the one that Figure 22 represents. This is because Figure 22 shows the arithmetic mean ot the fifty experiments, and it is affected by the experiments in which orders are not placed. Although the successful experiments they all have a different performance, two examples that can graphically represent their behaviour are shown in Figure 25 and Figure 26, where the simulation runs with lowest costs are represented.

Figure 25.Inventory level for simulation run 11

Figure 26. Inventory level for simulation run 12

Contrary to the five experiments with the highest costs, only 27,47% of the costs on average are due to the loss of sales whereas the 70,6% of the costs are due to the supply of the items. Besides, there is a significant variability in the order quantities, which vary from 0 to 285. It should be considered that in some cases having irregular ordering times and quantity orders could be a difficult issue to handle with suppliers.

To summarise, although at the beginning it seems that the distribution of the NN costs will overlap the distribution of the EOQ costs quite fast, it takes some iterations to really achieve it. In the beginning costs and inventory level are quite irregular: high costs and high inventory alternating with not ordering at all. Nevertheless, at some point, the experiments start to approach the EOQ level by lowering inventory and, consequently, the costs.

Although the NN does not surpass the EOQ model, a clear trend to approach it is proved (see Figure 21 and Figure 22). It might take more iterations more until the NN beats the EOQ model and has a stable behaviour, without stopping ordering from time to time.

5.2.2. Trial 2: var_rel = 0,85 - 1 / delivery_ontime = 1

In the second trial, var_rel is set at 0,85 and delivery_ontime is set at 1. This means that the goods ordered always arrive on time but with a varying delivery accuracy between a 85% and 100%. Thus, most of the times all the goods ordered are delivered, but sometimes exist some delivery failures. The results obtained after 50 EOQ experiments are shown in Figure 27.

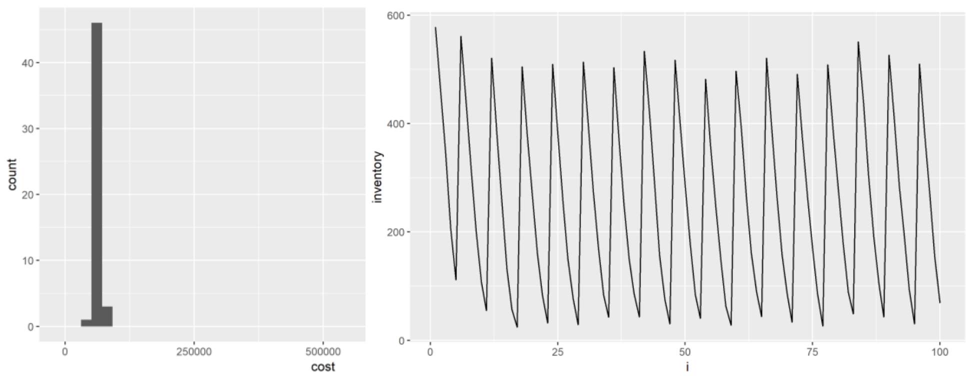


Figure 27. EOQ results for experiment 2

Two main differences when compared to the results obtained in the previous trial (see Figure 18), which has a 100% of reliability, can be noted. First, the distribution of costs is different. 92% of the experiments result in the same costs as in trial 1 but some of them result in higher costs and one of them in lower. The lower the *delivery_accuracy* is, the higher will be the costs. On the other hand, the change on the delivery reliability can also be noticed in the inventory graph. Since the demand is defined by the same distribution but less items are delivered, the inventory level has a downward trend instead. In order words, if less items are delivered but the number of items claimed by the customer is the same, the inventory will diminish as time goes by.

Table 8 shows the results for the EOQ experiments with the lowest and the highest costs. As well as in trial 1, the costs due to the loss of sales is what differentiates them. Whereas in simulation run 23 no costs due to this reason exist, they raise up to 18.622,5 in simulation 28, which means a 26,85% of its total costs.

Simulation run	Supply costs	Ordering costs	Lost sales costs	Inventory costs	Total costs
23	50.745	160	0	290,22	51.195,22
28	50.475	160	18.622,5	106,615	69.364,115

Table 8. Comparison between simulation run 23 and 28

In addition, the average delivery reliability for the 50 experiments and for the individual cases exposed above has been calculated. Table 9 shows the results, which indicate that delivery reliability and costs are not directly linked. Although in the simulation run with the highest costs the delivery reliability is slightly lower than in the simulation with the lowest costs, both are under the average of the 50 experiments, which is 92,57%. Hence, there are experiments with higher delivery reliability than in simulation run 23 that imply higher costs. Thus, contrary to expectations, a higher delivery reliability does not necessarily lead to lower costs.

	Simulation run 23	Simulation run 28	Totality of the experiments
Items ordered	11.088	11.088	554.400
Items delivered	10.149	10.095	513.209
% delivery reliability	91,53%	91,04%	92,57%

Table 9. % delivery reliability in EOQ experiments

The digital twin learning process is described and analysed below. After 50 iterations the inventory level is extremely high, reaching the 500000 units in stock, and therefore, the costs are considerably high too. During the following iterations, the inventory level and the distribution of costs start to slowly decrease approaching the EOQ model. However, it combines the decrease of inventory and costs with the behaviour adopted in iteration 100 of the previous trial (see Figure 20). This means not ordering at all. Even though there exist these episodes of instability, there is a clear downwards trend. After 275 iterations, the experiments result in the distribution of costs and inventory level shown in Figure 28.

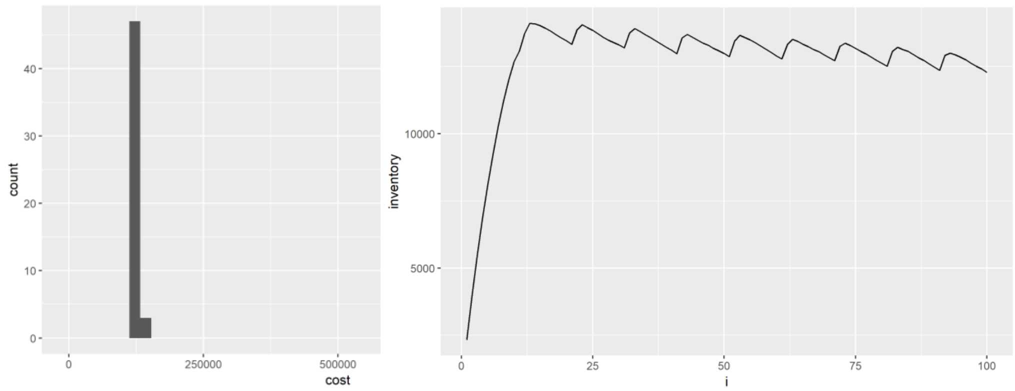


Figure 28. NN results for trial 2 (275 iterations)

Although the inventory level has decreased from 500.000 to less than 15.000, the distribution of costs does not overlap yet the one described by the EOQ model. This is due to the high amount of inventory ordered in the beginning of simulation. After a few periods, the inventory graph adopts a triangle-shaped form like the one described by the EOQ model and it becomes quite stable. However, the fact that at the beginning a huge number of items is ordered and obviously not sold, force the model to keep a high inventory and, consequently, the costs are still high.

Even though it might be reasonable that from now on the neural network would be a better decision maker and would keep improving, results in iteration 300 show that its behaviour is not stable yet. Suddenly, the digital twin decides again to place huge orders and the inventory level reaches the 80.000 units in stock. Nevertheless, as it is shown in Figure 29, the inventory level and costs sharply drop to low levels after 25 trainings more.

Comparing Figure 27 and Figure 29, it can be noticed that for the first time in this trial the distribution of costs obtained with the NN experimentation overlap the EOQ one. However, the distribution of costs is quite spread and only 5 out of 50 reach EOQ costs. Having such a big difference between the experiments influences the inventory level graph, since it represents the mean of all the experiments.

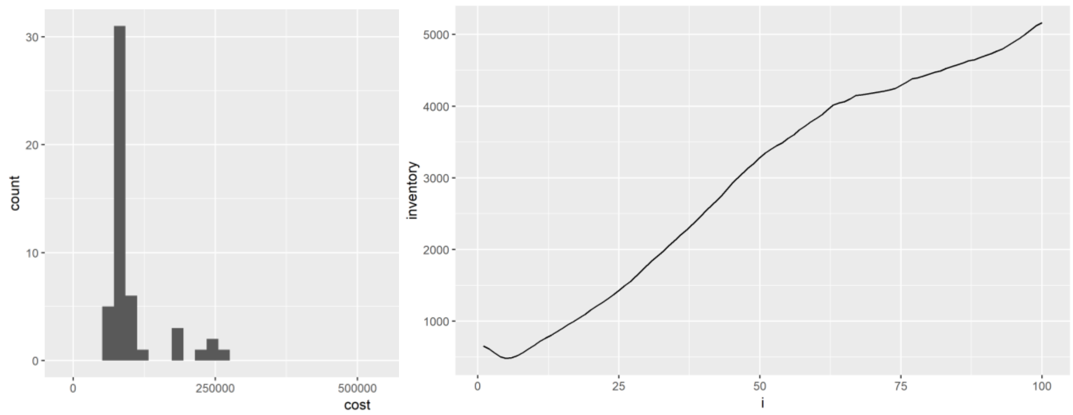


Figure 29. NN results for trial 2 (325 iterations)

In order to properly analyse the results, the 5 *successful* experiments along with the 5 experiments that lead to the highest costs have been deeply analysed and compared. As previously proved, the delivery reliability does not have a direct effect on the costs. Table 10 and Table 11 show the delivery reliability for the 5 experiments with the lowest and the highest costs, respectively. In both cases the delivery reliability is almost the same, and very close to the average considering the 50 experiments, which is 92,81%.

Simulation run	%delivery reliability
5	93,40%
14	92,19%
26	92,98%
28	92,61%
47	92,85%
MEAN	92,81%

Table 10. % delivery reliability for the 5 experiments with the lowest costs

Simulation run	%delivery reliability
17	92,23%
23	92,78%
41	--
42	93,00%
50	93,08%
MEAN	92,77%

Table 11.% delivery reliability for the 5 experiments with the highest costs

On the one hand, the behaviour adopted by 4 out of 5 of the experiments that lead to high costs is storing large quantities of items, reaching inventory levels of 25.000 or even 40.000 in some cases. An example of it is shown in Figure 30, where the inventory level for the experiment with the highest costs is represented. Although ordering too much is the most common reason of high costs in this trial, they can also be due to not ordering at all, as it is shown in Figure 31.

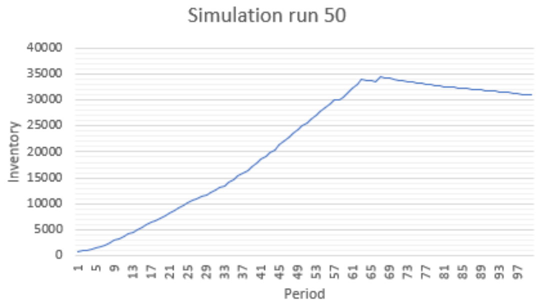


Figure 30. Inventory level for simulation run 50

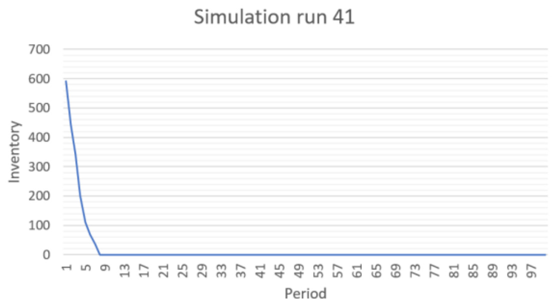


Figure 31. Inventory level for simulation run 41

On the other hand, Figure 32 shows the inventory graph for the experiment with the lowest costs. The inventory level in this experiment is much lower than in the experiments showed above. At first no orders are placed but from period 29 on, the digital twin starts to order again. As it can be noticed, the inventory graph has an irregular shape, this time not like the expected triangular shape that the EOQ model describes. This means that neither the order frequency nor the order size is stable. Besides, there is an accumulation of items at the end of the simulation.

However, the five experiments have not all the same behaviour as the one described above. The general trend, as shown in Figure 33, is to order more than needed until the half of the simulation, and at this point stop accumulating items in such a way that the inventory level at the end of the simulation is more less the same as at the beginning.

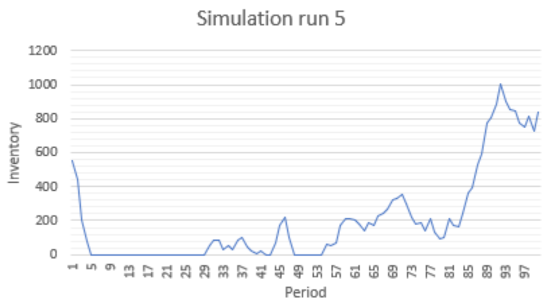


Figure 32.Inventory level for simulation run 5

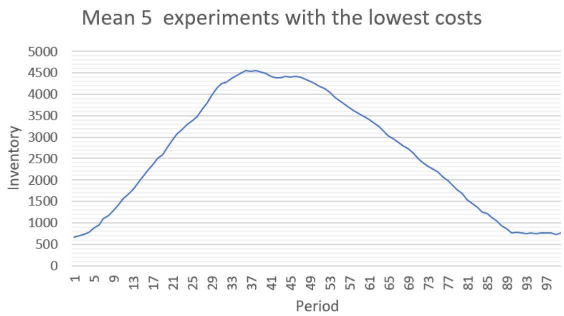


Figure 33. Mean of the 5 experiments with the lowest costs

In conclusion, when adding uncertainty to the simulation both the classical model and the neural network behave different than in the first experiment, where there was not. On the one hand, the uncertainty affects to the costs of the EOQ model, which are slightly higher in some of the experiments. Nevertheless, since the reliability is of an at least a 85% in every order, the effect is not very significant. Besides, the learning process of the digital twin is again not stable. In this trial, although it seems as if costs are progressively decreasing, the digital twin from time to time stops ordering or it suddenly orders a lot, as it happens in iteration 300. It is true than an approach to the classical model is proved, but only 10% of the experiments achieve it. As well as in the previous trial, it might take more iterations more until the NN beats the EOQ model and has a stable behaviour.

5.2.3. Trial 3: var_rel = 1 / delivery_ontime = 0,85

In the third trial, *var_rel* is set at 1 and *delivery_ontime* is set at 0,85. This means that the whole set of items ordered arrives, but not always on time. To keep it simple, it is supposed that when orders are delayed, they arrive the day after. The results obtained after 50 EOQ experiments are shown in Figure 34. The distribution of costs does not differ much from the one obtained in trial 1 (see Figure 18), where no delivery uncertainty was considered. However, the inventory graph is slightly different, mainly in the peaks, where the delivery delays are reflected.

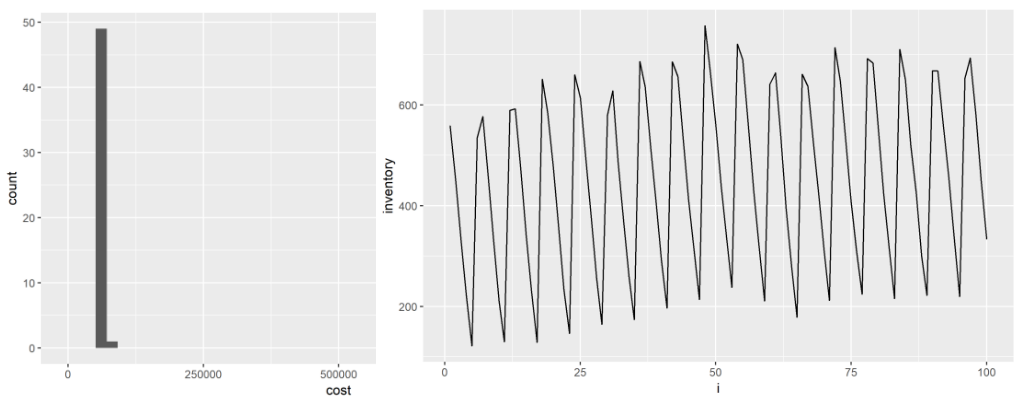


Figure 34. EOQ results for trial 3

Table 12 shows the results for the experiments with the lowest and the highest costs. Again, what makes the difference are the lost sales costs. Whereas in simulation run 11 the demand was fully satisfied in all the periods, costs due to the loss of sales rise to 19.177,5 in simulation 43, which means a 25,61% of its total costs.

Simulation run	Supply costs	Ordering costs	Lost sales costs	Inventory costs	Total costs
11	55.440	160	0	311,34	55.911,34
43	55.440	160	19.177,5	117,135	74.984,635

Table 12. Comparison between simulation run 11 and 43

In order to know to what extent a delay affects to the model, it has been calculated the percentage of lost sales costs that are due to a delay on the delivery of the goods ordered. Table 13 shows the results of this analysis, which indicate that an average of 20,05% of the lost sales costs are due to delay. In the case of simulation run 43, this number goes up to 31,52%.

	Simulation run 43	Totality of the experiments
Lost sales costs due to delay	6045	48.240
Lost sales costs without delay	13.132,5	192.405
Total lost sales costs	19.177,5	240.645

Table 13. Lost sales costs for simulation run 43 and for the totality of the experiments

In regard to the digital twin behaviour, the learning process is considerably unstable. After 50 iterations, the inventory level is huge. It reaches the level of 200.000 units and, consequently, the costs are high too. From that point both start to decrease, approaching the classical model. During the learning process, like in the previous trials, sometimes the digital twin decides not to order at all.

After 175 iterations, as shown in Figure 35, 14% of the experiments lead to the same costs as the EOQ model. For the first time in this trial the distribution of costs overlap the one obtained with the classical model experimentation (see Figure 34). However, the digital twin keeps ordering too much, being still far to achieve the optimal inventory level and costs.

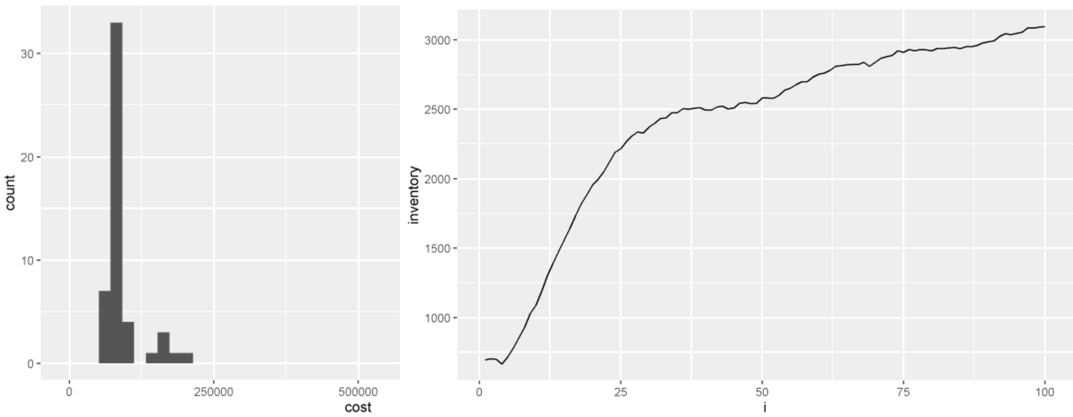


Figure 35. NN results for trial 3 (175 iterations)

Although it might seem that the digital twin is close to overcome the classical model, there is a sudden change on the learning process. Instead of keep decreasing, the inventory level and costs go up again, reaching the 200.000 units after 250 iterations. From that point, it starts to decline again. After 300 iterations, as it is shown in Figure 36, the distribution of costs overlaps again the one obtained with the EOQ experimentation. Nevertheless, in iteration 325 the digital twin decides not to order at all.

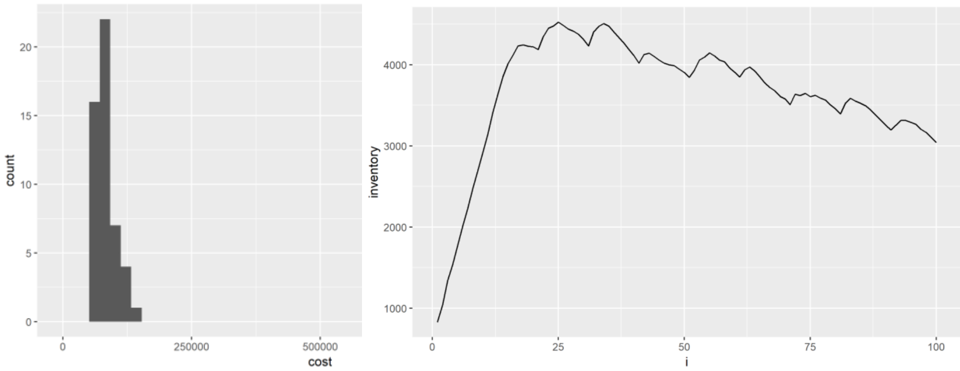


Figure 36. NN results for trial 3 (300 iterations)

The learning process of the digital twin is markedly unstable, with ups and downs that do not show a clear approach to the classical model. Nevertheless, it is true that after 300 iterations, the decisions of the digital twin are much better than at the beginning. In any case, more iterations would be needed to stabilise the decisions. For this trial, the results obtained after 300 iterations are the ones chosen to be analysed.

As the inventory graph shows in Figure 36 and as happens in most of the experiments of the three trials, first the digital twin decides to order huge quantities and accumulates inventory. After that, the inventory level adopts a triangle-shaped form like the one described by the EOQ model. Although a slightly downward trend is noticeable, there is still excessive stock.

The distribution of costs' graph shows that 16 out of 50 experiments lead to the same distribution of costs as the EOQ experiments. First, the whole set of experiments has been analysed. After that, these 16 *successful* experiments as well as the five experiments with the highest costs have been individually studied.

37 out of 50 experiments have been affected by a delivery delay. However, although the probability of receiving items late was set at 85%, all of them have been affected with a delivery reliability of at least a 90%. In the case of the 16 experiments with the lowest costs, 15 out of 16 experiments have received delayed orders but all of them with a delivery

reliability between a 93% and a 99%. In the case of the five experiments with the highest costs this range drops, and the delivery reliability is encountered between the 90% and 95%. Nevertheless, since the inventory levels are every high, the delays have not affected much to the costs because they do not cause the loss of sales.

In the case of the 13 experiments, only a 0,83% of the lost sales costs are due to a delay. In the experiments with the highest costs, the demand has been always fully satisfied and there are no costs due to the loss of sales.

Moreover, as well as in the previous trials, the experiments with the lowest costs and the ones with the highest ones have been individually plotted and are shown in Figure 37 and Figure 38. It is true that the mean represented in Figure 36 is useful to have a general idea of the digital twin's performance but since the experiments have a significant different behaviour, an individual representation of both cases helps to better interpret the results.

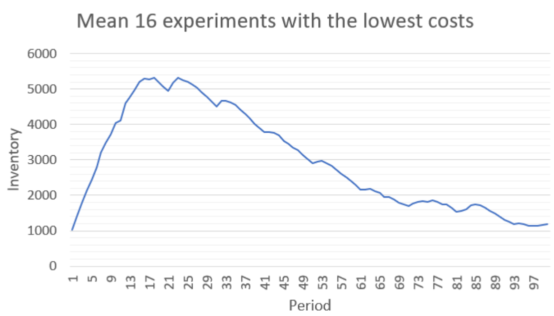


Figure 37. Mean of the 16 experiments with the lowest costs

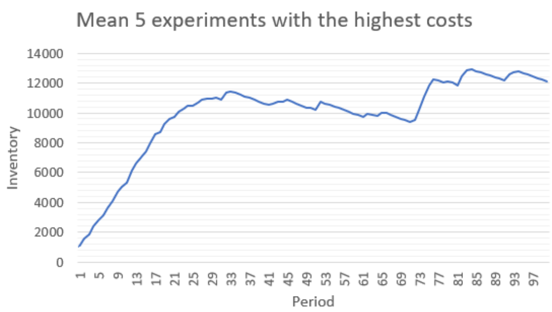


Figure 38. Mean of the 5 experiments with the highest costs

It can be noted that both cases tend to accumulate inventory at the beginning. However, in Figure 37 at some point the inventory level starts to decrease and its shape is closer the one adopted by the EOQ model. This means that a batch of items is ordered, demand is fulfilled during the next periods, and some periods further the next order is placed. Even though this fact can be also noticed in Figure 36, it is corrupted by the experiments with high costs, that contrary to that behaviour, keep ordering and storing items (see Figure 38).

Although the experiments represented in Figure 37 lead to the same costs as the classical model, the digital twin tends to accumulate inventory. This forces the company to have the capacity to store a big number of items and may imply a waste of resources. In addition, the percentage of lost sales on average for the EOQ experiments is 5,31% whereas it reaches the 14,08% for the digital twin's 16 experiments with the lowest costs. Although it does not affect the costs, decreasing lost might be a matter of interest for many companies.

To sum up, the learning process of the digital twin in the third trial is considerably unstable. Although an approach to the classical model is achieved, after 325 iterations the digital twin is still not able to make good decisions. It tends either to order too many items in the beginning or not ordering at all. A change in these two aspects might help to improve the decision-making process.

On the other hand, after 300 iterations only 32% of the experiments lead to the same results as the EOQ model. In addition, analysing these *successful* experiments, it has been noticed that they require more storage capacity and they cause on average a higher percentage of lost sales than the classical model.

5.3. Implications

Once conducted the three trials and analysed the results, it is stated that uncertainty affects the performance of the inventory control. The EOQ model does not consider demand variability, and this is what affects the most to the EOQ experiments. In the three trials, the difference between the experiment with the lowest costs and the one with the highest costs are the costs due to the loss of sales. At best, all the demand is fulfilled. In the worst case, there is an increase in the costs due to this reason that goes up to 16,06%, 36,38% and 25, 61% in each of the trials, respectively. It is important to note that the percentage of costs due to the loss of sales is higher in the second and third trials, where delivery accuracy is modified.

When modifying the delivery reliability, although the distribution of costs and the inventory level is only slightly affected, some changes can be appreciated. In the second trial, where all the orders arrive with a reliability between 85% and 100%, 8% of the experiments lead to a different distribution of costs than the first trial. Since fewer items are received, the supply costs are lower but the lost sales costs much higher. In the third trial, the distribution of costs is not affected but it is the inventory graph. In addition, 20% of the lost sales costs are due to the delay in the orders.

Regarding the learning process of the digital twin, some general insights can be highlighted. The classical model has not been beaten by the digital twin in any of the trials. However, an improvement can be noted in its behaviour when comparing its decisions in the first iterations with the ones after 300 iterations. In the beginning, the distribution of costs is much higher than the one obtained with the EOQ experimentation. After around 300 iterations, the digital twin in the three trials overlaps the EOQ distribution of costs curve. In the first and third trial, 30% and 32% of the experiments achieve it, respectively. In the second trial, this number is a bit lower, achieving success only in 10% of the experiments.

The learning process of the digital twin is unstable in the three trials. In all of them, there is a downward trend of costs and inventory, alternated with episodes of ordering from time to time or ordering huge quantities. Besides, in most of the experiments, the digital twin tends to order too much at the beginning and stabilise afterwards. This may be due to the fact that the initial inventory level given to the neural network in the first iteration is always calculated with the EOQ formula.

In addition, it can be pointed out that all the experiments in the three trials start having a triangle-shaped form like the one described by the EOQ model but after several iterations, they all evolve to very unstable forms. This means that both order quantities and order frequency are quite variable. This should be considered because it could be an obstacle when dealing with suppliers.

In conclusion, this experimentation proves that the digital twin can learn from historical data and, in some of the cases, can achieve the same results as the EOQ model. However, some changes would be needed in order to bring it to real-life use. First implementations that could be done are improving the accuracy of the NN and check whether the information given to the NN could be expanded. In addition, more learning iterations could be run in order to let more time to the model to stabilise.

6. Conclusions

This chapter includes the conclusions of the thesis. Considering both the literature research and the experiments conducted, an answer to the questions raised in chapter 1.3 is given. Besides, it also presents the limitations of the project and some recommendations for future research.

6.1. Summary

The aim of the thesis was to test whether a digital twin can be a better decision-maker than a classical model and know which are the challenges that need to be faced when implementing it. First, thanks to extensive research work, the concepts of supply chain management and digital twin have been introduced. After that, three different trials challenging the limitations of the EOQ model have been carried out.

It is very important for companies to consider that supply chain management is not only logistics but also product development, marketing, sales and production, among others. And all of them need to be managed. To do so and to take competitive advantage, data analytics is essential. It should also be considered that, although firms generate enormous quantities of data, not all of them are useful. The important thing is to know how to disregard useless data. However, this might require a significant investment that not all companies are willing to make.

Besides, because of taking part in a supply chain, firms might be affected by uncertainty. Although they cannot avoid them, what it is in their power is to try to reduce the consequences caused by them, thus reducing overall inefficiencies in the whole supply chain.

This thesis proposes a process-based digital twin to try to solve one of the main issues that companies should deal with in SCM, which is inventory control. An object-based digital twin can improve an object lifecycle, but it cannot make its own decisions because it is not able to consider external factors. Consequently, it needs a human to make the final decision. Contrary to that, a process-based digital twin, since all the process is represented, the dependency on the human factor disappears. In addition, the fact of receiving real-time data allows the digital to keep continuously learning and improving.

Implementing digital twins in companies is advisable since they can have a significant contribution to business performance. They can combine the data collected from the process with data from the company in order to look for new opportunities for saving costs, improving quality and increasing efficiency. Thus, they can make predictions and future decisions. What is more, they allow testing new procedures without the need of creating prototypes, being able to make as many changes as necessary without wasting resources.

In this thesis, a digital twin simulation is tested to prove whether it can beat a classical inventory control model, the EOQ. Three trials are carried out to test the learning capacity of a reinforcement learning application. Any factor that impacts on the performance of the inventory control and that allows to create different scenarios would be useful for a comparison with a classical model. However, since this project is limited to only a few trials, it is considered that the best factors to challenge a classical model are the ones that overcome the classical model limitations. Because of this, two factors in relation to delivery accuracy are chosen.

Results of the experimentation are not as successful as expected. It is proved that the digital twin simulation can learn from historical data and its behaviour improve after several training iterations. In two of the three trials, 30% of the experiments result in the same distribution of costs as the EOQ model, whereas in the last one only 10% of them achieve it. This means that the distribution of costs' curve of the digital twin overlaps the one of the EOQ model, but it is not able to overcome the classical model.

Although the digital twin does not beat the EOQ model, it is considered that the aim of the thesis is achieved. Despite the project limitations, the learning process of the digital twin simulation is proved. When implementing a process-digital twin some challenges need to be faced. Regarding technical issues, in this project, the accuracy of the model has been diminished in order to avoid the simulation runs to be stuck. In real implementations, companies might need to deal, for instance, with the need of installing large amounts of sensors, investing in disregarding useless data or in security systems.

Besides, it is possible that the implementation of a digital twin would not be accepted in the beginning. On the one hand, organisations are usually reluctant in investing in technologies where profit is not ensured yet. On the other hand, some companies or suppliers might be afraid of sharing information with the digital twin information system.

Creating a process-digital twin is somewhat challenging. However, it is proven that its implementation is worthy and that companies that have implemented it are already taking profit of it. A process-based digital twin can also be implemented in other fields apart from the SCM, such as health care, manufacturing, automation or even in smart cities.

6.2. Limitations

This thesis discusses the importance of data analytics in supply chain management and proposes the digital twin as a possible technology to do it. The ideal target would be to simulate a whole supply chain, considering all the entities taking part in it and all the factors that affect their performance. Since the time available to develop this thesis was limited to one semester and due to the degree of complexity that this simulation would require, only a first approach to it has been developed. Thus, the simulation is only based on one single entity of the supply chain and it represents one of the many issues that companies must face when trying to manage the SC: inventory control.

Apart from the shortage of time to develop this thesis, the running time needed for the reinforcement learning process is another factor that limited the scope of the project. First iterations were rapidly run, but the time needed to run them increased as the number of iterations was higher. In order to speed up the process and being able to perform the experiments, the threshold for the partial derivatives of the neural network was changed. Although this accelerated the process, it deteriorated the accuracy of the model. Besides, despite the improvements, after some time they became slow again. This was a limitation on both the number of trainings in each experiment and the number of experiments carried out.

The number of trainings was set at 325, which was considered enough to analyse the learning process of the digital twin and to determine an approach to the classical model. However, some more iterations would be needed to overcome the EOQ model. On the other hand, the fact of only being able to perform three experiments impeded the study of other interesting scenarios, such as adding more parameters to the simulations or combine the existing ones to analyse how their interaction affects the model.

In addition, not having real data constrained the experimentation too. Some random data were created in a logical way and given to the model to train. Even though this was enough for a first experimentation, it is always better to have real data in order to start with a solid basis.

6.3. Future research

As previously mentioned in chapter 6.2, this thesis is a limited approach to analyse a very widetopic as it is supply chain management. It has been proved that a digital twin can learn from experience and give similar solutions to the ones obtained with classical models. However, further investigation can be carried out in several areas.

Firstly, the aim of this thesis could be extended. It would be interesting to run more iterations to analyse which is the best distribution of costs that the digital twin can achieve. Besides, more experiments could be performed changing some parameters such as costs, demand or demand standard deviation and try several combinations of them. Also, more factors that disobey the EOQ model limitations could be added to the experimentation.

Secondly, other classical models could be challenged. The EOQ model has been chosen because of its simplicity but it could also be appealing to analyse other classical inventory control models and see whether an approach to is feasible using a digital twin.

On the other hand, a second step towards SCM could be taken by considering other issues beyond inventory control. As exposed in chapter 6.3, digital twins can be also used in many other fields. Some examples would be the prediction of machine failures, maintenance, customised production, etc.

Finally, the simulation of a whole supply chain could be modelled. This would be the result of the compilation of all the ideas exposed above. However, this would be a large-scale project and would require lots of resources in terms of time, knowledge and computing power, among others. In order to successfully achieve it, it is recommended to first model each of the entities taking part in the SC and proceed to their integration afterwards. In addition, more research in the field of digital twins and the connection between the digital and the real world would be required.

7. Reference list

ADA3DS (2018) *Digital twins, 3 Casos de Éxito de su Implementación,* 21 May. Available at: https://ada3ds.com/blog-noticias-interes/digital-twins-3-casos-de-exito-de-su-implementacion/ (Accessed: 16 June 2019).

Andersen, J. (2019) *Where Do Digital Twins Fit In?,* 17 May. Available at: https://www.industryweek.com/technology-and-iiot/where-do-digital-twins-fit (Accessed: 17 May 2019).

Automation, S. (2019) *Q&A: Transforming SAGE's manufacturing process with industry 4.0*. Available at: https://www.sageautomation.com/blog/qa-transforming-sages-manufacturing-process-with-industry-4.0 (Accessed: 17 May 2019).

Banker, S. (2018) *Digital Twins Support Supply Chain Optimization*. Available at: https://www.forbes.com/sites/stevebanker/2018/11/22/digital-twins-support-supply-chain-optimization/ (Accessed: 16 June 2019).

Barrera, D. (2016) 'Modelos Determinísticos y Probabilísticos'.

Caudill, M. (1989) 'Neural Network Primer: Part I'.

Cooper, M.C., Lambert, D.M. and Pagh, J.D. (1997) 'Supply Chain Management: More Than a New Name for Logistics', *The International Journal of Logistics Management*, 8(1), pp. 1–14. doi: 10.1108/09574099710805556.

Dahiya, A. (2019) *Introduction to Reinforcement Learning!,* 12 May. Available at: https://www.xadahiya.me/blog/2018/06/26/RL-Intro/ (Accessed: 12 June 2019).

Deloitte (2012) 'Supply Chain Analytics'.

Deloitte Unity Press (2017) 'DUP_Industry-4.0_digital-twin-technology'.

Dertat, A. (2017) *Applied Deep Learning - Part 1: Artificial Neural Networks*. Available at: https://towardsdatascience.com/applied-deep-learning-part-1-artificial-neural-networks-d7834f67a4f6 (Accessed: 11 June 2019).

Domingos, P. (2016) 'A Few Useful Things to Know about Machine Learning'.

Dormehl, L. (2019) *What is an artificial neural network? Here's everything you need to know | Digital Trends,* 6 January. Available at: https://www.digitaltrends.com/cool-tech/what-is-an-artificial-neural-network/ (Accessed: 11 June 2019).

Ellis, S. (2019) *Top 10 Predictions for Worldwide Supply Chains in 2019,* 23 May. Available at: https://www.mhlnews.com/global-supply-chain/top-10-predictions-worldwide-supply-chains-2019 (Accessed: 23 May 2019).

Gartner (2017) *Top Trends in the Gartner Hype Cycle for Emerging Technologies, 2017*. Available at: https://www.gartner.com/smarterwithgartner/top-trends-in-the-gartner-hype-cycle-for-emerging-technologies-2017/ (Accessed: 20 May 2019).

Gartner (2018) *5 Trends Emerge in the Gartner Hype Cycle for Emerging Technologies, 2018*. Available at: https://www.gartner.com/smarterwithgartner/5-trends-emerge-in-gartner-hype-cycle-for-emerging-technologies-2018/ (Accessed: 20 May 2019).

Gartner (2019) *Gartner Survey Reveals Digital Twins Are Entering Mainstream Use*. Available at: https://www.gartner.com/en/newsroom/press-releases/2019-02-20-gartner-survey-reveals-digital-twins-are-entering-mai (Accessed: 15 June 2019).

General Electric (2018) ‘GE Digital Twin’.

Gii Finance Network (2016) *An introduction to supply chain analytics – and why it’s so important*. Available at: https://giifinance.com/blog/an-introduction-to-supply-chain-analytics-and-why-its-so-important/ (Accessed: 17 May 2019).

Gomez, G. (2019) *Introducción al Aprendizaje por Refuerzo – Planeta Chatbot : todo sobre los Chatbots, Voicebots e Inteligencia Artificial*. Available at: https://planetachatbot.com /introducci%C3%B3n-al-aprendizaje-por-refuerzo-f910d669d077 (Accessed: 31 May 2019).

Gravelle, E. (2018) *Shield AI Fundamentals: On Reinforcement Learning*. Available at: https://www.shield.ai/content/2018/11/20/shield-ai-fundamentals-on-reinforcement-learning (Accessed: 31 May 2019).

Hanneman, S. (2017) ‘The Procedd Digital Twin: A Step towardsoperational excellence.’.

Hernández, N. (2018) *‘Digital Twin’: los objetos físicos buscan a su gemelo digital*. Available at: https://www.nobbot.com/negocios/digital-twin-los-objetos-fisicos-buscan-a-su-gemelo-digital/ (Accessed: 16 June 2019).

IBM Knowledge Center (2014) *Nodo Red neuronal*. Available at: https://www.ibm.com/ support/knowledgecenter/es/SS3RA7_sub/modeler_mainhelp_client_ddita/ clementine/trainnetnode_general.html (Accessed: 11 June 2019).

Jose David Pinilla Manrique (2011) *Desventajas de los modelos EOQ y LEP,* 29 August. Available at: http://ingjox.blogspot.com/2011/05/desventajas-de-los-modelos-eoq-y-lep.html (Accessed: 29 May 2019).

Kaeser (2018) *Sigma Smart Air: Maintenance with pinpoint precision – KAESER KOMPRESSOREN*. Available at: https://www.kaeser.com/int-en/company/press/press-releases/m-sigma-smart-air.aspx (Accessed: 16 June 2019).

Kent, G. (2018) *¿Qué Es Machine Learning? [Guía Completa Para Principiantes]*. Available at: https://blog.adext.com/machine-learning-guia-completa/ (Accessed: 12 June 2019).

Kitain, L. (2018) *Digital Twin—The New age of Manufacturing – Data Driven Investor – Medium*. Available at: https://medium.com/datadriveninvestor/digital-twin-the-new-age-of-manufacturing-d964eeba3313 (Accessed: 17 May 2019).

Koch, R. (2016) *High-Quality Data: The Best Fuel for Supply Chain Change | Logistics Viewpoints*. Available at: https://logisticsviewpoints.com/2016/04/07/high-quality-data-the-best-fuel-for-supply-chain-change/ (Accessed: 17 May 2019).

Kumar, R. (2016) 'Economic Order Quantity (EOQ) Model'. Global Journal of Finance and Economic Management, 5, pp 1-5.

La Londe, B.J. (1997) 'Supply Chain Management: Myth or Reality? Supply Chain Management Review'.

Larson, Q. (2018) *An introduction to Reinforcement Learning*. Available at: https://www.freecodecamp.org/news/an-introduction-to-reinforcement-learning-4339519de419/ (Accessed: 12 June 2019).

Logistics & Materials Handling Blog (2012) *What is the Bullwhip Effect? Understanding the concept & definition*. Available at: https://www.aalhysterforklifts.com.au/index.php/about/blog-post/what_is_the_bullwhip_effect_understanding_the_concept_definition (Accessed: 17 May 2019).

Mahanta, J. (2017) *Introduction to Neural Networks, Advantages and Applications*. Available at: https://towardsdatascience.com/introduction-to-neural-networks-advantages-and-applications-96851bd1a207 (Accessed: 11 June 2019).

Markets and Markets (2018) *Digital Twin Market worth 15.66 Billion USD by 2023*. Available at: https://www.marketsandmarkets.com/PressReleases/digital-twin.asp (Accessed: 18 June 2019).

Marr, B. (2017) *What Is Digital Twin Technology - And Why Is It So Important?* Available at: https://www.forbes.com/sites/bernardmarr/2017/03/06/what-is-digital-twin-technology-and-why-is-it-so-important/ (Accessed: 17 May 2019).

McMahon, A. (2018) *Reinforcement Learning in the Supply Chain*. Available at: https://medium.com/streamba-data/reinforcement-learning-in-the-supply-chain-

2fd56ab59c2e (Accessed: 30 May 2019).

Mentzer, J.T., DeWitt, W., Keebler, J.S., Min, S., Nix, N.W., Smith, C.D. and Zacharia, Z.G. (2001) 'Journal of Business Logistics: Defining Supply Chain Management', 22.

Merriam-Webster (2019) *Definition of INVENTORY CONTROL,* 29 May. Available at: https://www.merriam-webster.com/dictionary/inventory%20control (Accessed: 29 May 2019).

Mikell, M. and Clark, J. (2018) *Cheat sheet: What is Digital Twin? Internet of Things blog*. Available at: https://www.ibm.com/blogs/internet-of-things/iot-cheat-sheet-digital-twin/ (Accessed: 17 May 2019).

Morales, E. (2016) 'Aprendizaje por Refuerzo'.

Mussomeli, A., Meeker, B., Shepley, S. and Schatsky, D. (2018) 'Deloitte Insights. Expecting Digital Twins'.

NC State University (2017) *What is Supply Chain Management (SCM)?* Available at: https://scm.ncsu.edu/scm-articles/article/what-is-supply-chain-management-scm (Accessed: 16 May 2019).

Ognjanovski, G. (2019) *Everything you need to know about Neural Networks and Backpropagation — Machine Learning Made Easy…* Available at: https://towardsdatascience.com/everything-you-need-to-know-about-neural-networks-and-backpropagation-machine-learning-made-easy-e5285bc2be3a (Accessed: 11 June 2019).

Patil, P., Shrotri, A.P. and Dandekar, A.R. (2012) 'Management of Uncertainty in Supply Chain'.

Pontius, N. (2016) *Data: A Key Ingredient in Supply Chain Optimization*. Available at: https://www2.camcode.com/asset-tags/using-data-in-supply-chain-optimization/ (Accessed: 17 May 2019).

Rouse, M. (2016) *What is unsupervised learning? - Definition from WhatIs.com*. Available at: https://whatis.techtarget.com/definition/unsupervised-learning (Accessed: 12 June 2019).

Sachin Agarwal (2014) 'Economic Order Quantity: A Review'.

Schmid, U. (2005) 'Reinforcement Learning: Bamberg Universität'.

Schoenherr, T. and Speier-Pero, C. (2015) 'Data Science, Predictive Analytics, and Big

Data in Supply Chain Management: Current State and Future Potential', *Journal of Business Logistics*, 36(1), pp. 120–132. doi: 10.1111/jbl.12082.

Shetty, B. (2018) *Supervised Machine Learning: Classification – Towards Data Science*. Available at: https://towardsdatascience.com/supervised-machine-learning-classification-5e685fe18a6d (Accessed: 12 June 2019).

Silver, E.A., Pyke, D. and Peterson, R. (1998) 'Inventory Management and Production Scheduling'.

Simangunsong, E., Hendry, L.C. and Stevenson, M. (2012) 'Supply-chain uncertainty: a review and theoretical foundation for future research', *International Journal of Production Research*, 50(16), pp. 4493–4523. doi: 10.1080/00207543.2011.613864.

Stevens, G.C. (1989) 'Integration of the Supply Chain: International Journal of PhysicalDistribution and Logistics Management', 19, pp. 3–8.

Thomas Ohnemus (2018) *Digital Twin Excellence: Two Shining Examples*. Available at: https://www.digitalistmag.com/iot/2018/06/14/digital-twin-excellence-2-shining-examples-06175901 (Accessed: 16 June 2019).

Trkman, P., McCormack, K., Oliveira, M.P.V. de and Ladeira, M.B. (2010) 'The impact of business analytics on supply chain performance', *Decision Support Systems*, 49(3), pp. 318–327. doi: 10.1016/j.dss.2010.03.007.

Ujjwalkarn (2016) *A Quick Introduction to Neural Networks*. Available at: https:// ujjwalkarn.me/2016/08/09/quick-intro-neural-networks/ (Accessed: 11 June 2019).

Unity consulting & Innovation (2018) *Digital Twin*. Available at: https://www.unity.de/en/ services/digital-twin/ (Accessed: 17 June 2019).

Wang, G., Gunasekaran, A., Ngai, E.W.T. and Papadopoulos, T. (2016) 'Big data analytics in logistics and supply chain management: Certain investigations for research and applications', *International Journal of Production Economics*, 176, pp. 98–110. doi: 10.1016/j.ijpe.2016.03.014.

Wideskog, M. (2018) 'Wärtsilä Hybrid Solutions - AVL Large Engine'.

Annex

A. Code: Digital twin simulation

This section shows snippets of the modifications done to the original code.

Trial 2. Modificacions to include delivery variability

Neural Network decision engine

```
NN_calc_1 = function(NN, Demand, h_cost_mu, h_demand_mu, h_demand_sd,
inventory, h_delivery_accuracy_mu){

  # create predictions of relevant decision options for orders
  decision_options = data.table(h_cost_mu = h_cost_mu, # Left hand side
is
                                # (conzinued) name in data frame, right
hand side is varible of function
                                h_demand_mu = h_demand_mu,
                                h_demand_sd = h_demand_sd,
                                inventory = inventory,
                                orders = seq(0,Demand/2,1),
                                h_delivery_accuracy_mu =
h_delivery_accuracy_mu)

  # normalize values
  decision_options = Norm_additional(x = c("h_cost_mu", "h_demand_mu",
"h_demand_sd", "orders", "inventory", "h_delivery_accuracy_mu"),
                                     dt = decision_options, trial = 2)
```

Simulation in generic form

```
# Simulation function with various parameters as input
Sim_function = function(sim_type = "EOQ", Demand = 12000, Demand_sd =
600, periods_in_year = 100, return_periods = 100,
                   ordering_cost = 10, interest = 0.1, Cost_purchase = 5,
Cost_lostSales = 7.5,
                   var_rel = NULL, delivery_ontime = 1, h_length = 100,
NN = NULL){
```

```
# 3. Simulation wit for-loop
  var_rel = runif(return_periods, 0.85,1)
  for(i in 1:return_periods){
        # 3.1 Update inventory
    # 3.1.1 add late arrivals due to bad on-time delivery performance
from previous day
    # and set temporary variable to 0 (default)
    incoming_goods = one_day_late
    one_day_late = 0

    # 3.1.2 Reorder based on Simulation type
    switch(sim_type,

           # 3.1.2 Case A: EOQ Model
           "EOQ" = {}, # orders have been pre-calculated already above
           "RAND" = {orders[i] = Rand_calc(Demand, Demand_sd,
periods_in_year)},
           "NN" = {orders[i] = NN_calc_1(NN, Demand, mean(h_cost),
mean(h_demand), sd(h_demand), inventory, mean(h_delivery_accuracy))}
           )

    # 3.1.3 apply delivery reliability on reorder process
    if(orders[i] > 0){
      if(runif(1, min = 0, max = 1) > delivery_ontime){
        one_day_late = one_day_late + round(orders[i] * var_rel[i])
        }else{
          incoming_goods = incoming_goods + round(orders[i] * var_rel[i])
        }
      }
```

```
# 3.5 recording all data in a list format
    Sim_data[[i]] = data.frame(i,
                               d_demand = d_demand[i], # otherwise the
variable is called d_demand.i. in the data frame
                               sales,
                               lost_sales,
                               inventory,
                               orders = orders[i],
                               incoming_goods,
                               var_rel = var_rel[i],
                               one_day_late,
                               c_supply,
                               c_ordering,
                               c_lost_sales,
                               c_inventory,
                               cost,
                               h_demand_mu = mean(h_demand),
                               h_demand_sd = sd(h_demand),
                               h_delivery_accuracy_mu =
mean(h_delivery_accuracy),
                               h_delivery_ontime_mu =
mean(h_delivery_ontime),
                               h_cost_mu = mean(h_cost))
```

Train neural network

```
# Normalization of the variable
train_init_data = Norm_initial(x = c("h_cost_mu", "h_demand_mu",
"h_demand_sd", "orders", "inventory",
"cost_mu_prediction","h_delivery_accuracy_mu"),
                              dt = train_init_data, trial = 2)


# initial training
train_init_NN = neuralnet(cost_mu_prediction ~ h_cost_mu + h_demand_mu +
h_demand_sd + orders + inventory + h_delivery_accuracy_mu,
                          data = train_init_data,
                          hidden = c(5, 3, 1),
                          act.fct = "logistic",
                          threshold = 0.005,
                          lifesign = "full",
                          lifesign.step = 500)
```

Reinforcement learning

```
  # (2) de-normalize
  train_init_data = Norm_de(x = c("h_cost_mu", "h_demand_mu",
"h_demand_sd", "orders", "inventory","h_delivery_accuracy_mu",
"cost_mu_prediction"),
                              dt = train_init_data, trial = 2)

  # (3) add simulation results
  train_init_data = rbindlist(list(train_init_data, temp), fill = FALSE)

  # (4) normalize data again
  train_init_data = Norm_initial(x = c("h_cost_mu", "h_demand_mu",
"h_demand_sd", "orders", "inventory",
"h_delivery_accuracy_mu","cost_mu_prediction"),
                              dt = train_init_data, trial = 2)

  # (5) re-train neural network
  not_done = TRUE
  thres = 0.01
  while(not_done) {
    candidate = neuralnet(cost_mu_prediction ~ h_cost_mu + h_demand_mu +
h_demand_sd + orders + inventory + h_delivery_accuracy_mu,
                          data = train_init_data,
                          hidden = c(5, 3, 1),
                          act.fct = "logistic",
                          threshold = thres,
                          lifesign = "full",
                          lifesign.step = 2000,
                          startweights = train_init_NN$weights) # this is
the re-training!!!
```

Trial 3. Modificacions to include delivery on time

Neural Network decision engine

```
NN_calc_1 = function(NN, Demand, h_cost_mu, h_demand_mu, h_demand_sd, 
inventory, h_delivery_ontime_mu){

  # create predictions of relevant decision options for orders
  decision_options = data.table(h_cost_mu = h_cost_mu, # Left hand side 
is
                                # (conzinued) name in data frame, right 
hand side is varible of function
                                h_demand_mu = h_demand_mu,
                                h_demand_sd = h_demand_sd,
                                inventory = inventory,
                                orders = seq(0,Demand/2,1),
                                h_delivery_ontime_mu = 
h_delivery_ontime_mu)

  # normalize values
  decision_options = Norm_additional(x = c("h_cost_mu", "h_demand_mu", 
"h_demand_sd", "orders", "inventory", "h_delivery_ontime_mu"),
                                     dt = decision_options, trial = 3)
```

Simulation in generic form

```
# NEEDED TO RUN, IF aNY SIMULATION IS SUPPOSED TO RUN

# Simulation function with various parameters as input
Sim_function = function(sim_type = "EOQ", Demand = 12000, Demand_sd = 
600, periods_in_year = 100, return_periods = 100,
                   ordering_cost = 10, interest = 0.1, Cost_purchase = 5, 
Cost_lostSales = 7.5,
                   var_rel = NULL, delivery_ontime = 0.85, h_length = 
100, NN = NULL){
```

```
# 3. Simulation wit for-loop
  var_rel = runif(return_periods, 1,1)
  for(i in 1:return_periods){
        # 3.1 Update inventory
    # 3.1.1 add late arrivals due to bad on-time delivery performance
from previous day
    # and set temporary variable to 0 (default)
    incoming_goods = one_day_late
    one_day_late = 0

    # 3.1.2 Reorder based on Simulation type
    switch(sim_type,

           # 3.1.2 Case A: EOQ Model
           "EOQ" = {}, # orders have been pre-calculated already above
           "RAND" = {orders[i] = Rand_calc(Demand, Demand_sd,
periods_in_year)},
           "NN" = {orders[i] = NN_calc_1(NN, Demand, mean(h_cost),
mean(h_demand), sd(h_demand), inventory, mean(h_delivery_ontime))}
           )

    # 3.1.3 apply delivery reliability on reorder process
    if(orders[i] > 0){
      if(runif(1, min = 0, max = 1) > delivery_ontime){
        one_day_late = one_day_late + round(orders[i] * var_rel[i])
        }else{
          incoming_goods = incoming_goods + round(orders[i] * var_rel[i])
        }
      }
```

```
# 3.5 recording all data in a list format
    Sim_data[[i]] = data.frame(i,
                               d_demand = d_demand[i], # otherwise the
variable is called d_demand.i. in the data frame
                               sales,
                               lost_sales,
                               inventory,
                               orders = orders[i],
                               incoming_goods,
                               var_rel = var_rel[i],
                               one_day_late,
                               c_supply,
                               c_ordering,
                               c_lost_sales,
                               c_inventory,
                               cost,
                               h_demand_mu = mean(h_demand),
                               h_demand_sd = sd(h_demand),
                               h_delivery_accuracy_mu =
mean(h_delivery_accuracy),
                               h_delivery_ontime_mu =
mean(h_delivery_ontime),
                               h_cost_mu = mean(h_cost))
```

Train neural network

```
# Normalization of the variable
train_init_data = Norm_initial(x = c("h_cost_mu", "h_demand_mu",
"h_demand_sd", "orders", "inventory",
"cost_mu_prediction","h_delivery_ontime_mu"),
                               dt = train_init_data, trial = 3)


# initial training
train_init_NN = neuralnet(cost_mu_prediction ~ h_cost_mu + h_demand_mu +
h_demand_sd + orders + inventory + h_delivery_ontime_mu,
                          data = train_init_data,
                          hidden = c(5, 3, 1),
                          act.fct = "logistic",
                          threshold = 0.005,
                          lifesign = "full",
                          lifesign.step = 500)
```

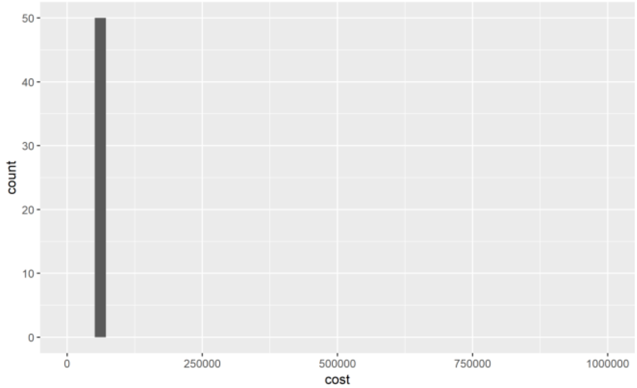
Reinforcement learning

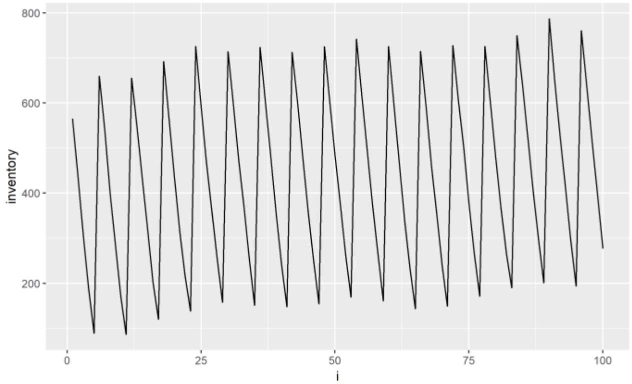
```
  # (2) de-normalize
  train_init_data = Norm_de(x = c("h_cost_mu", "h_demand_mu",
"h_demand_sd", "orders", "inventory","h_delivery_ontime_mu",
"cost_mu_prediction"),
                               dt = train_init_data, trial = 3)

  # (3) add simulation results
  train_init_data = rbindlist(list(train_init_data, temp), fill = FALSE)

  # (4) normalize data again
  train_init_data = Norm_initial(x = c("h_cost_mu", "h_demand_mu",
"h_demand_sd", "orders", "inventory",
"h_delivery_ontime_mu","cost_mu_prediction"),
                               dt = train_init_data, trial = 3)

  # (5) re-train neural network
  not_done = TRUE
  thres = 0.01
  while(not_done) {
    candidate = neuralnet(cost_mu_prediction ~ h_cost_mu + h_demand_mu +
h_demand_sd + orders + inventory + h_delivery_ontime_mu,
                          data = train_init_data,
                          hidden = c(5, 3, 1),
                          act.fct = "logistic",
                          threshold = thres,
                          lifesign = "full",
                          lifesign.step = 2000,
                          startweights = train_init_NN$weights) # this is
the re-training!!!
```

B. Experimentation

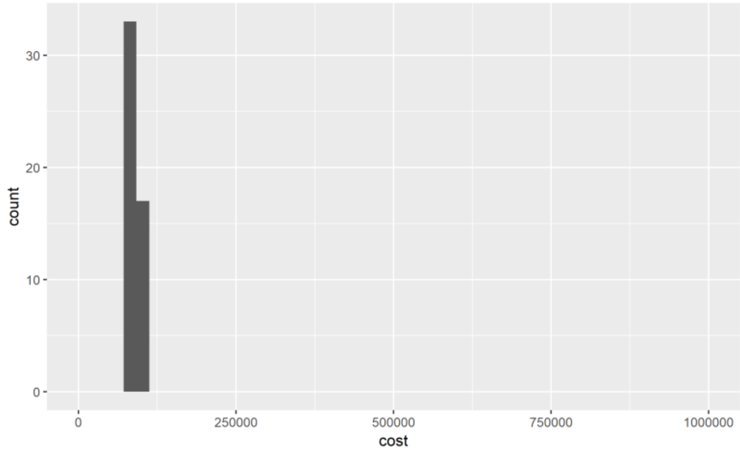
Trial 1: var_rel = 1 / delivery_ontime = 1

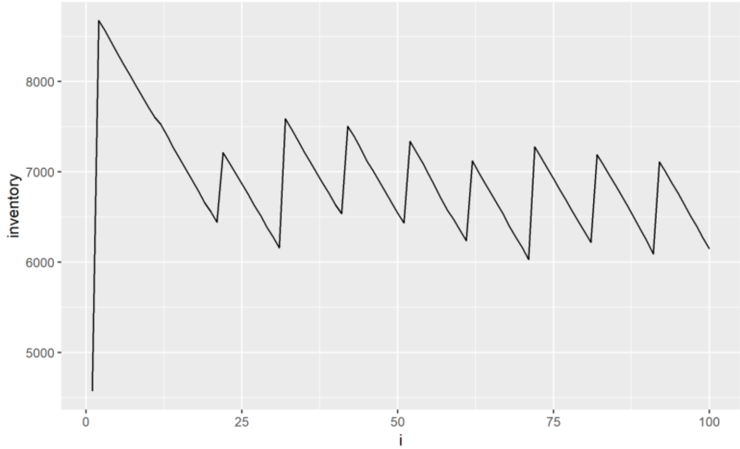
50 EOQ experiments



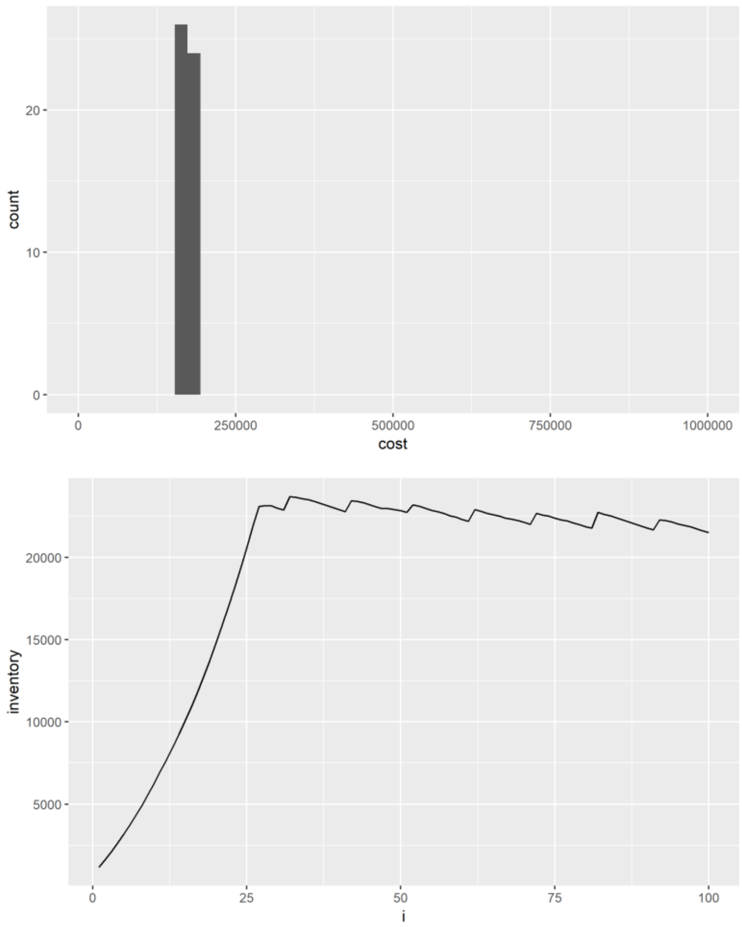


50 iterations NN

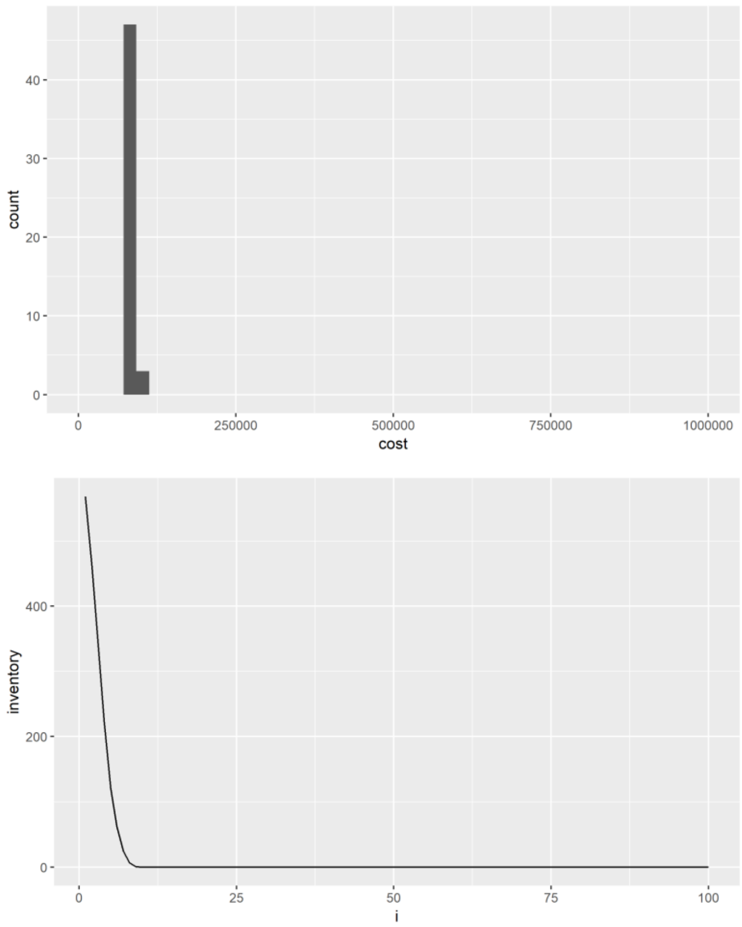




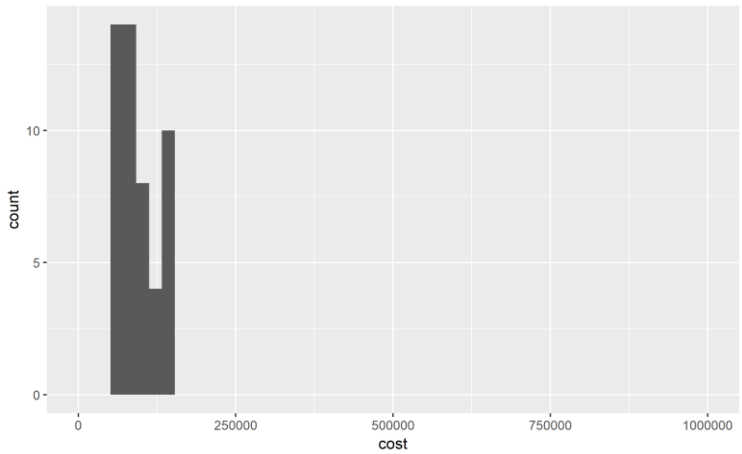
75 iterations NN

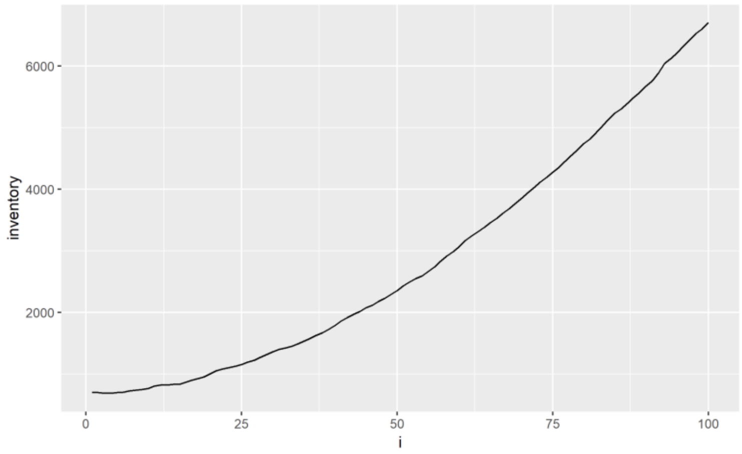


100 iterations NN

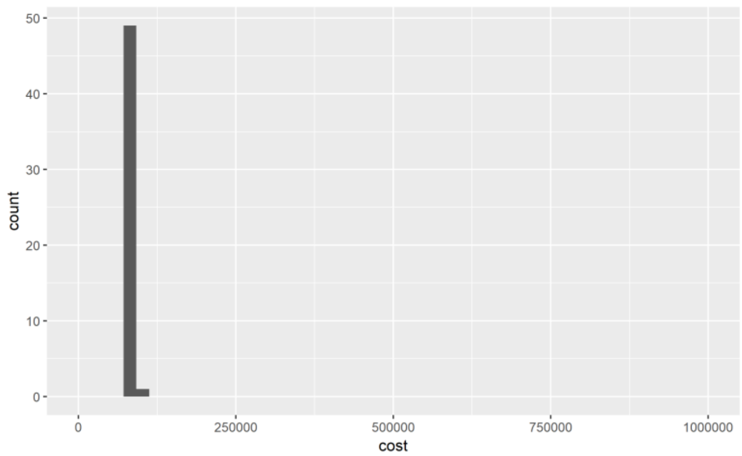


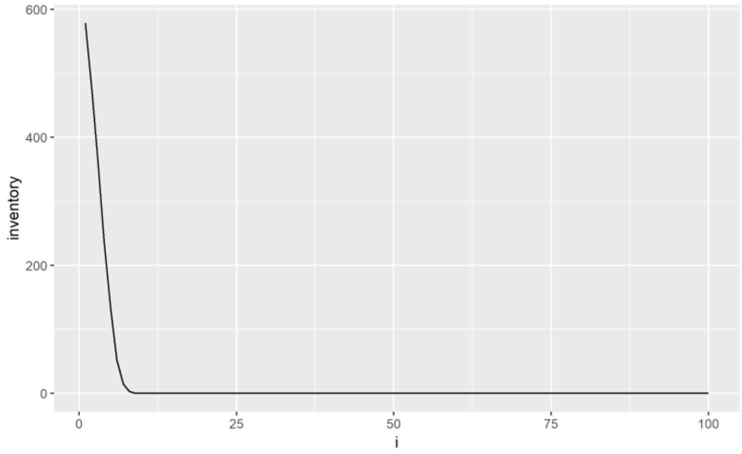
125 iterations NN



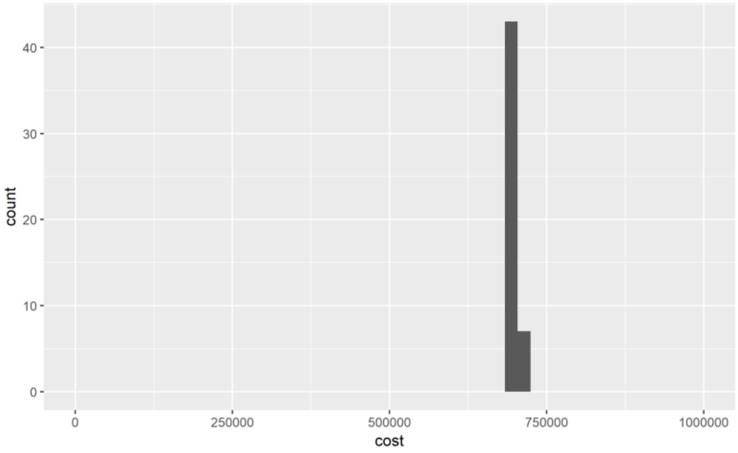


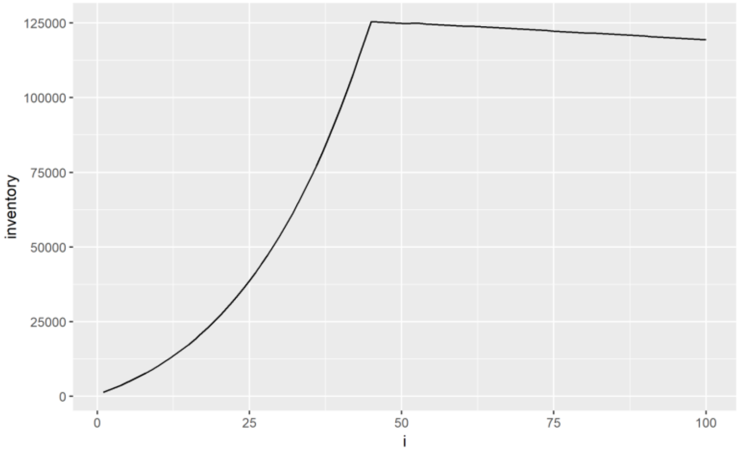
150 iterations NN



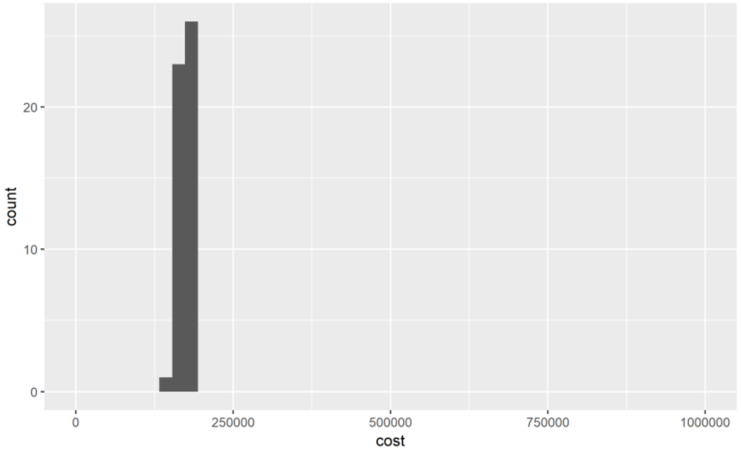


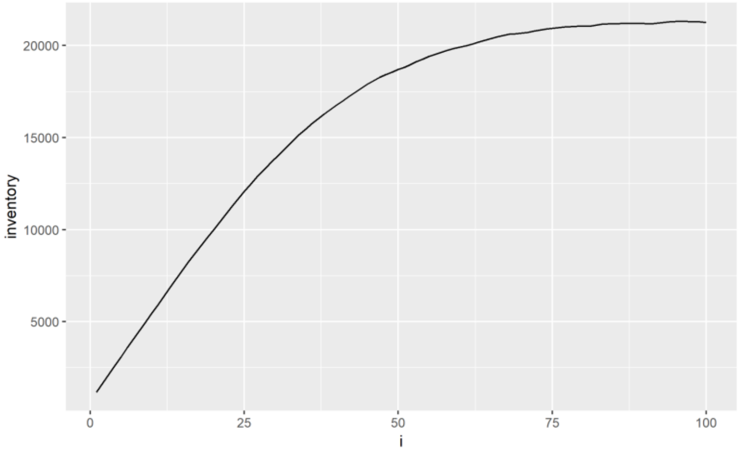
175 iterations NN



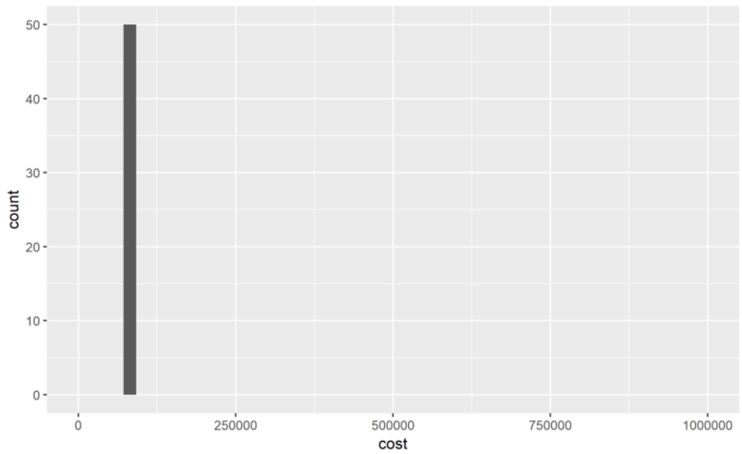


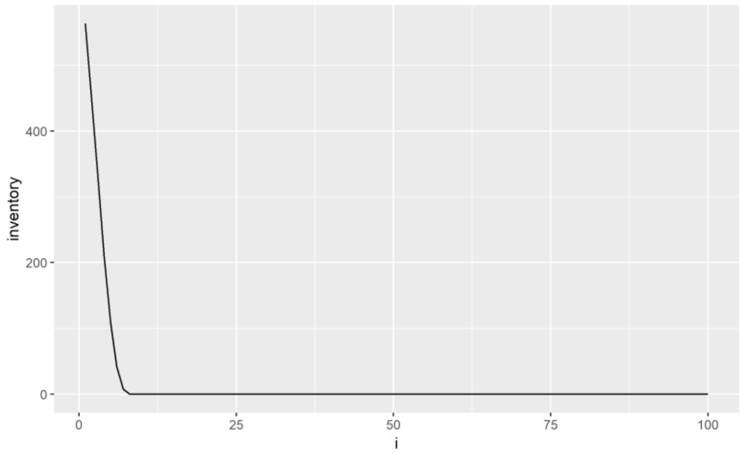
200 iterations NN



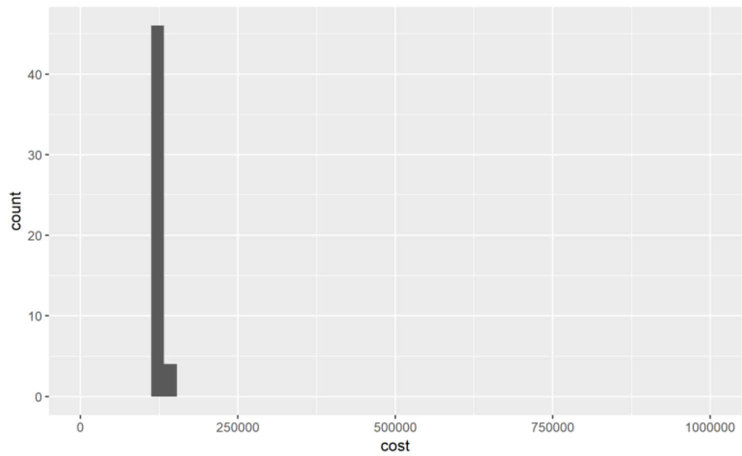


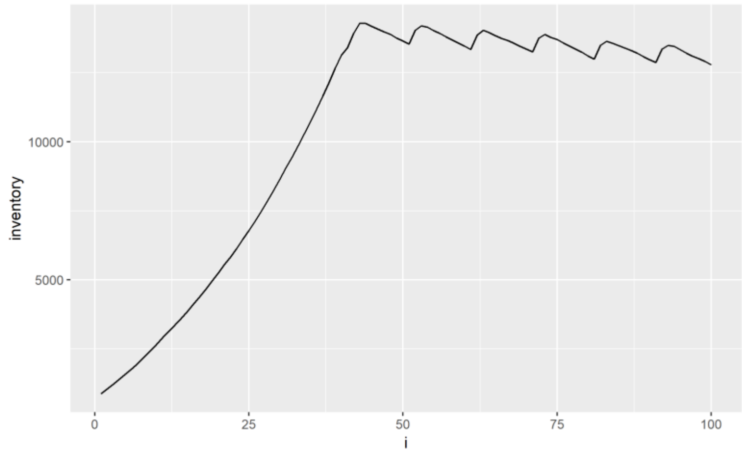
225 iterations NN



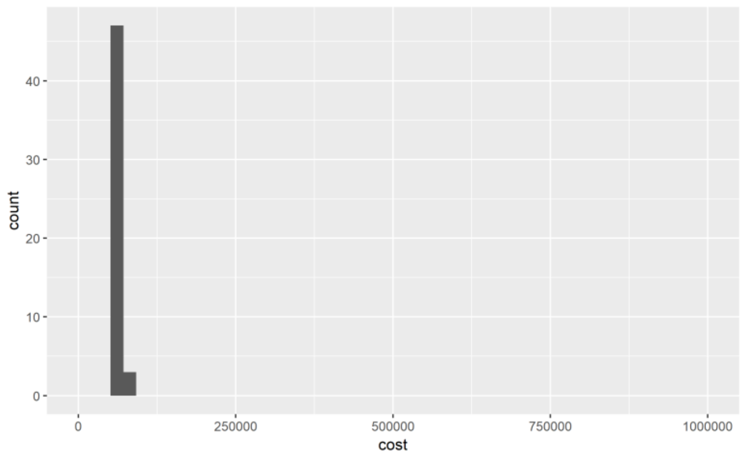


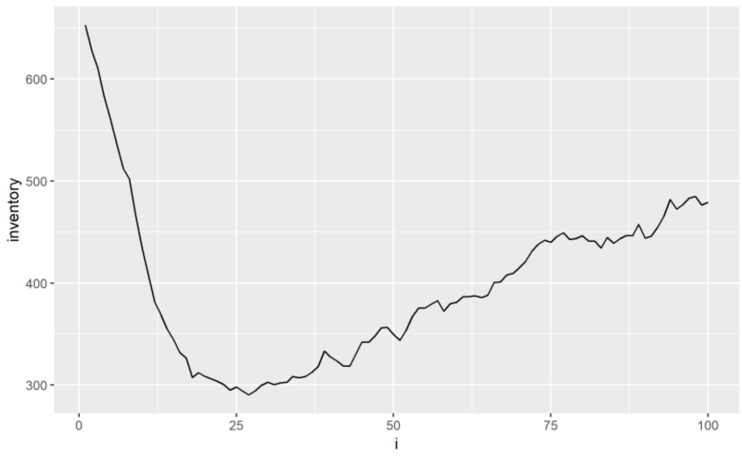
250 iterations NN



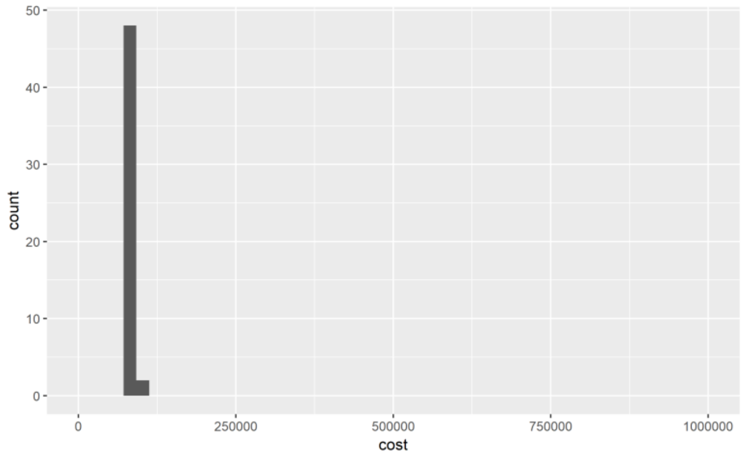


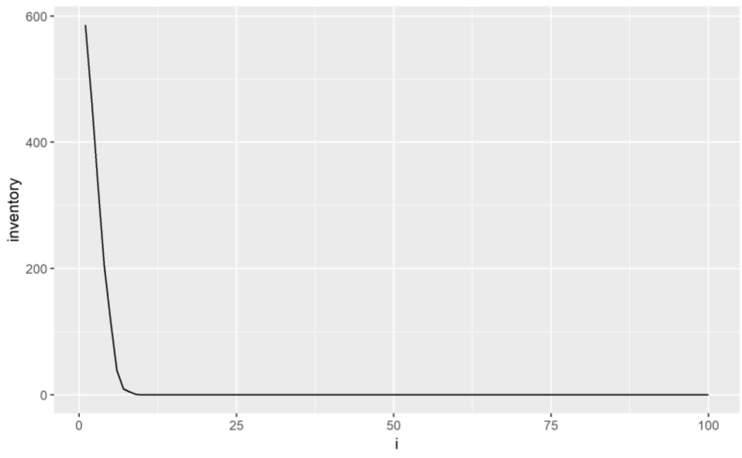
275 iterations NN



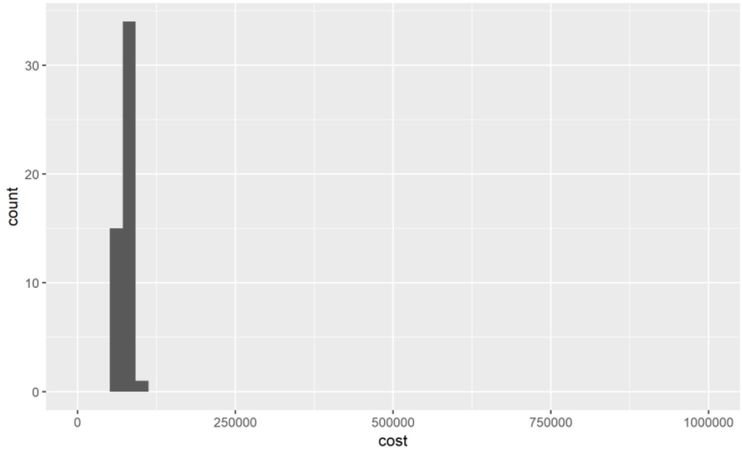


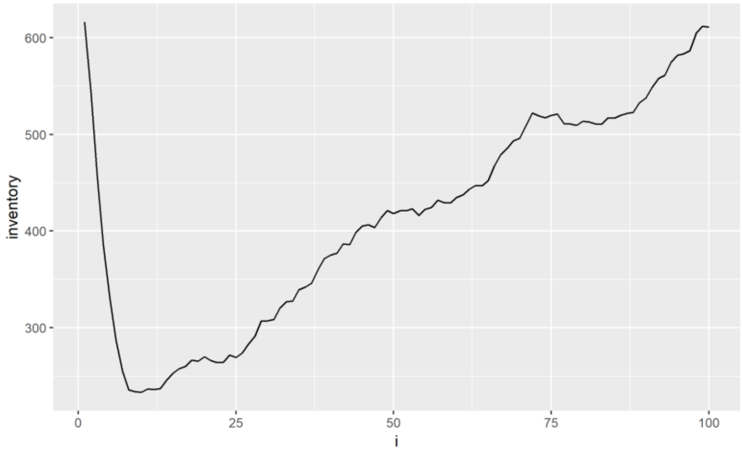
300 iterations NN





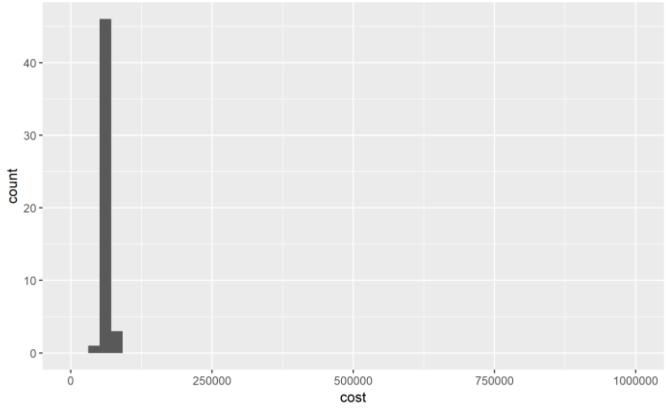
325 iterations NN

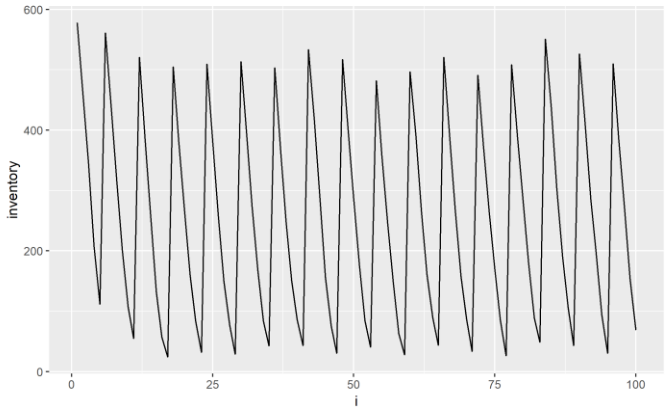




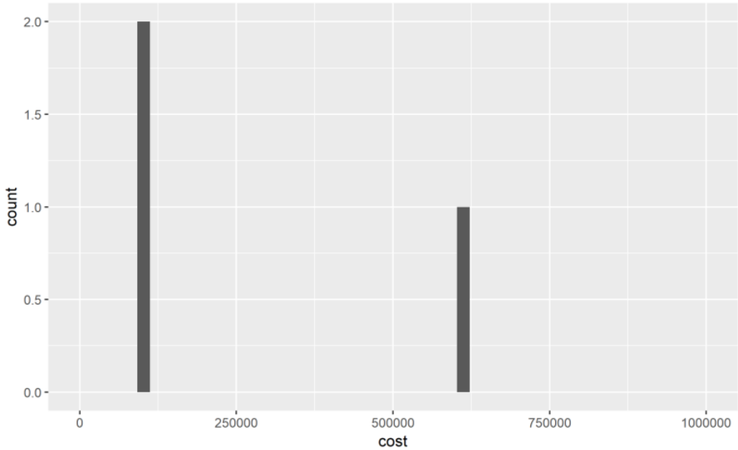
Trial 2: var_rel = 0,85 - 1 / delivery_ontime = 1

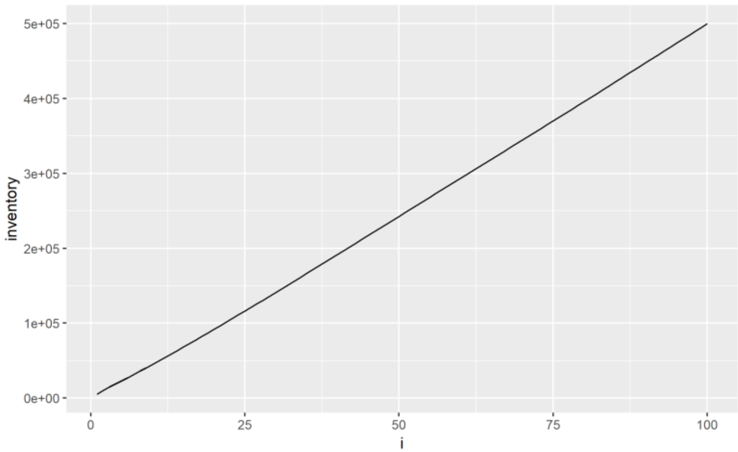
50 EOQ experiments



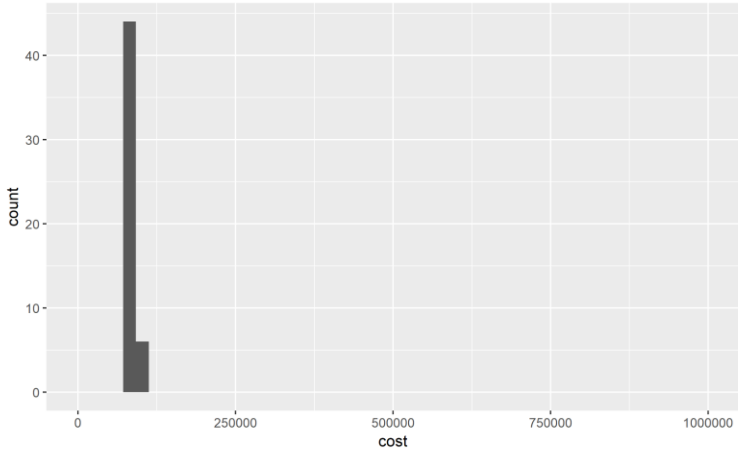


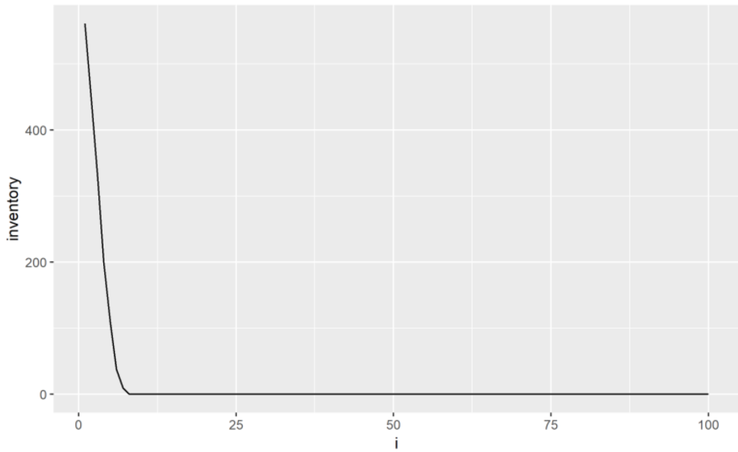
50 iterations NN



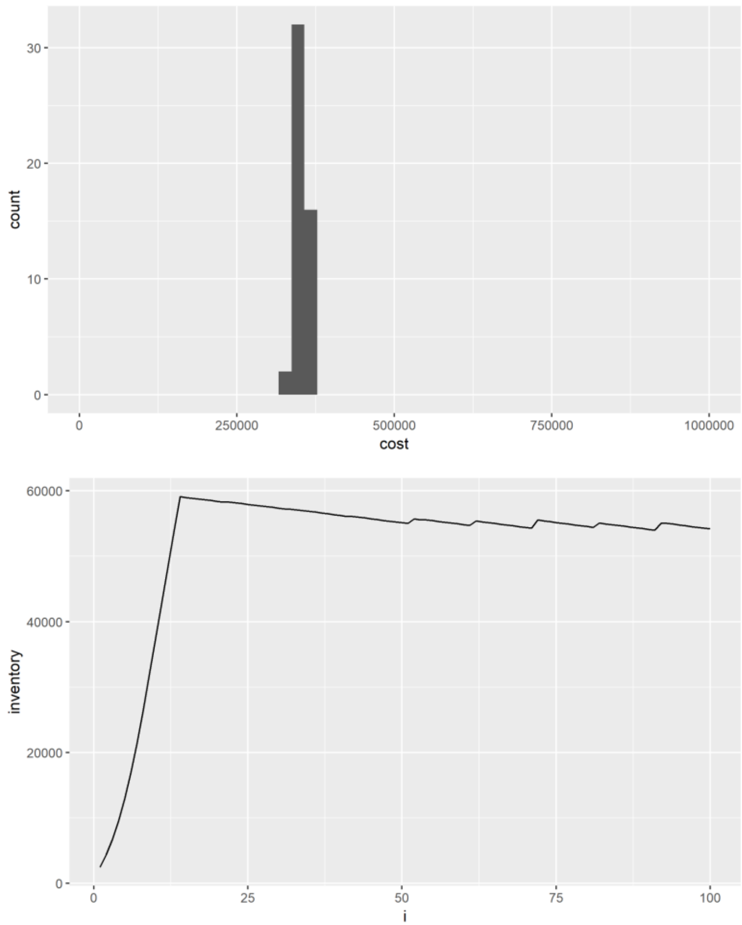


75 iterations NN

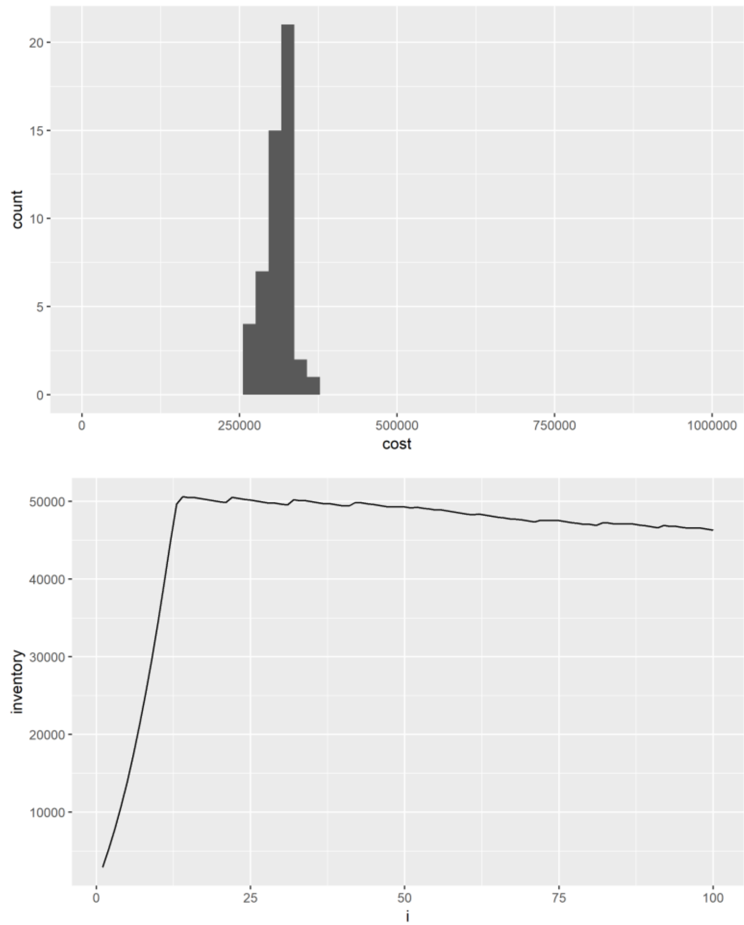




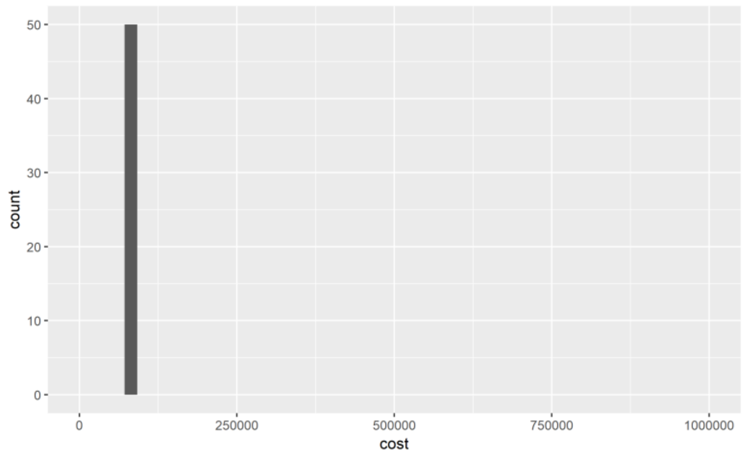
100 iterations NN

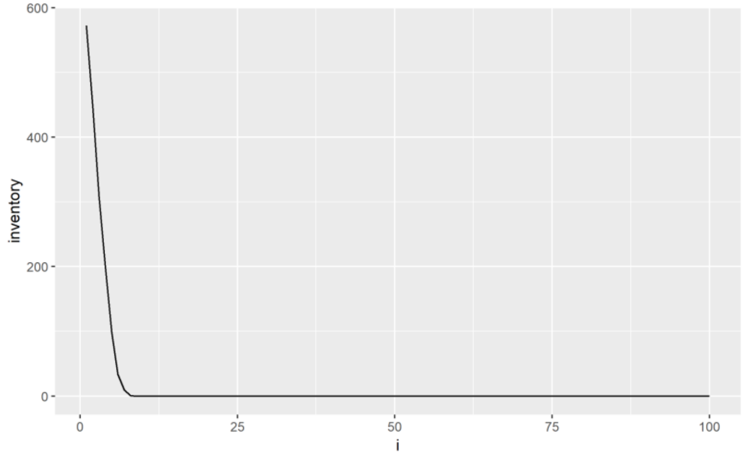


125 iterations NN

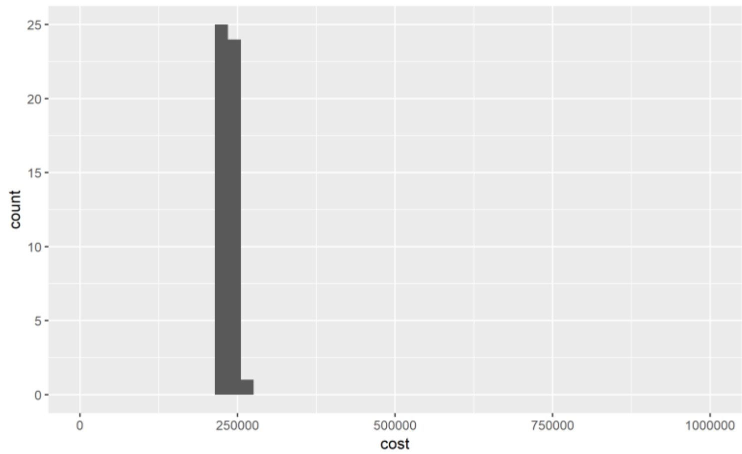


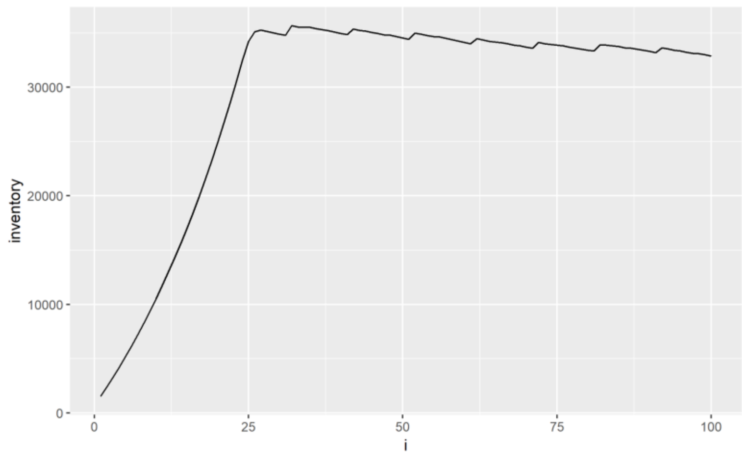
150 iterations NN



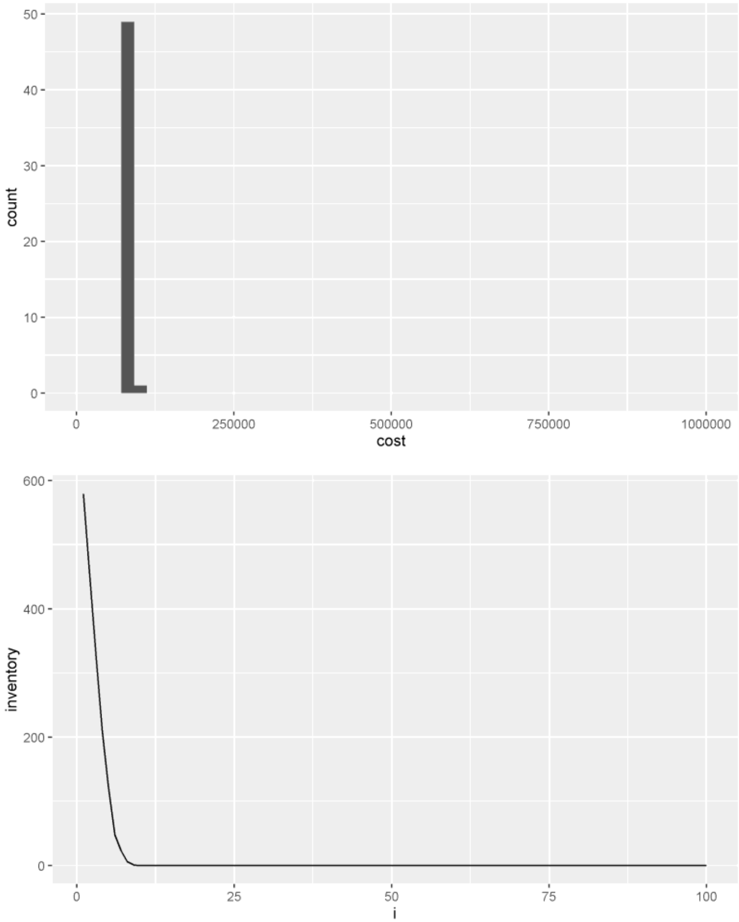


175 iterations NN

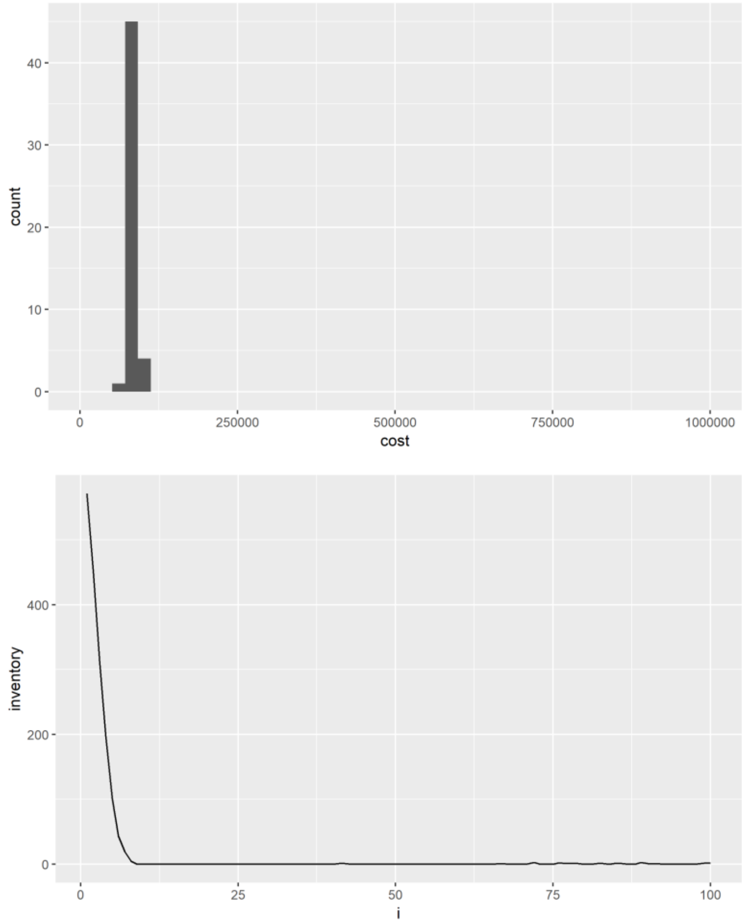




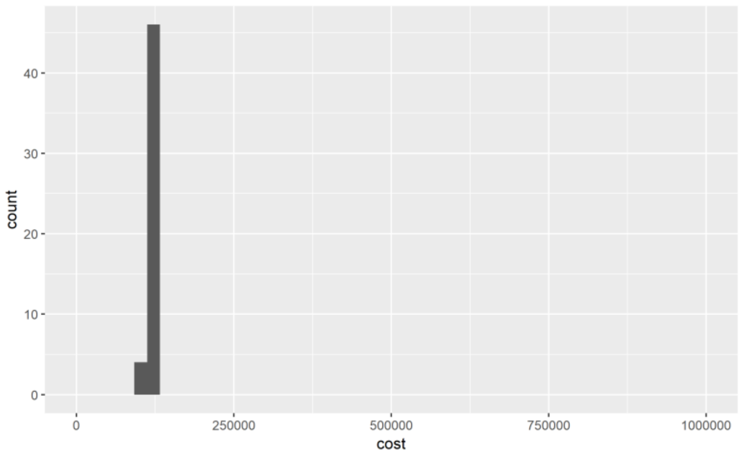
200 iterations NN

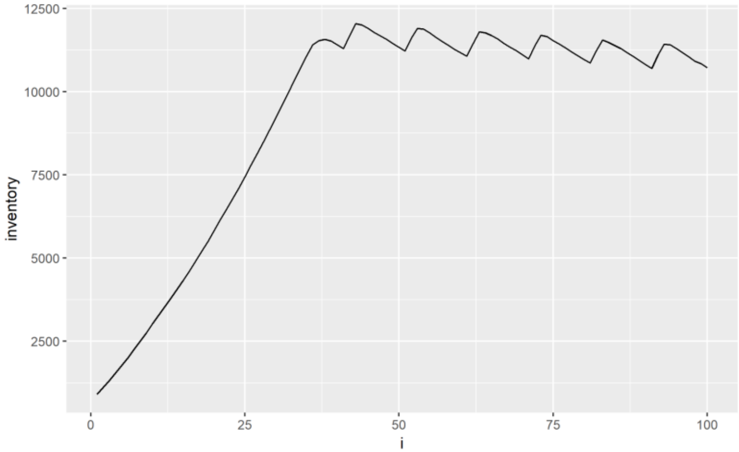


225 iterations NN

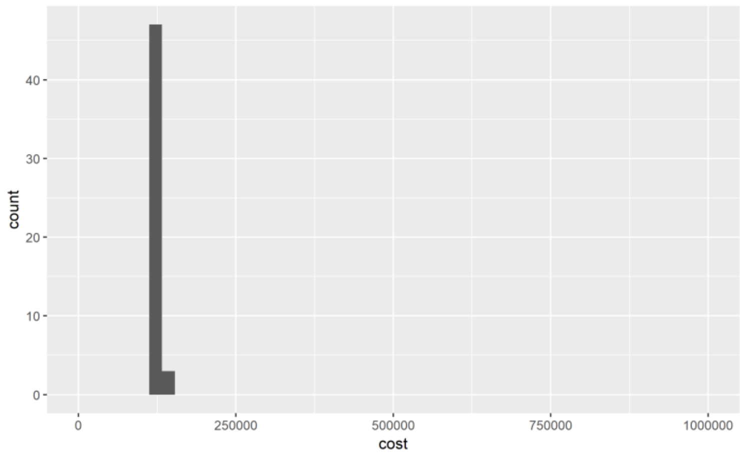


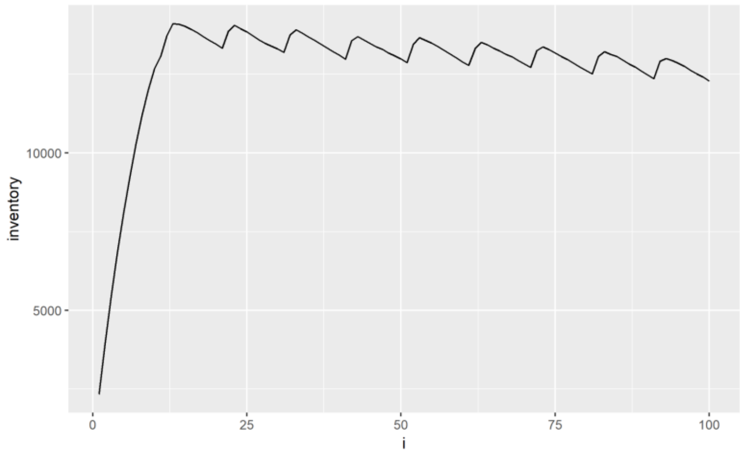
250 iterations NN



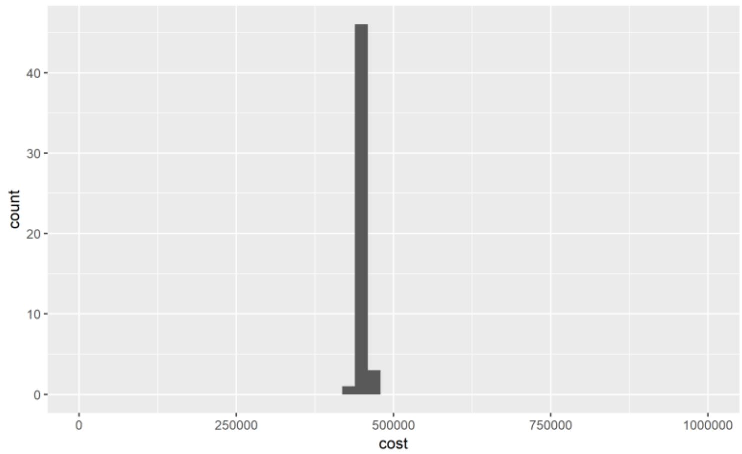


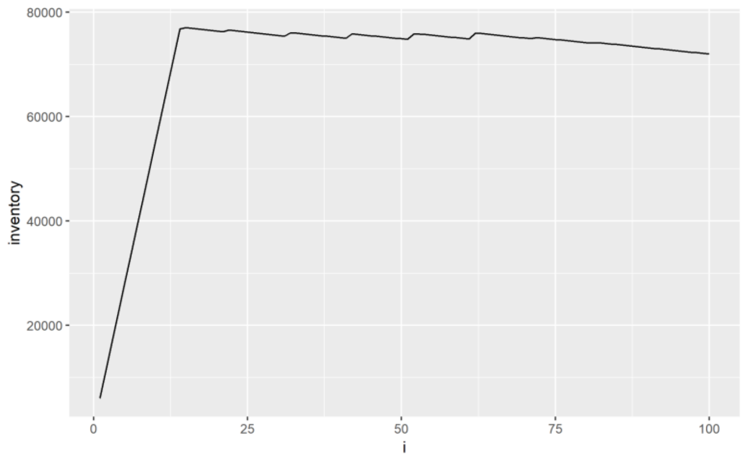
275 iterations NN



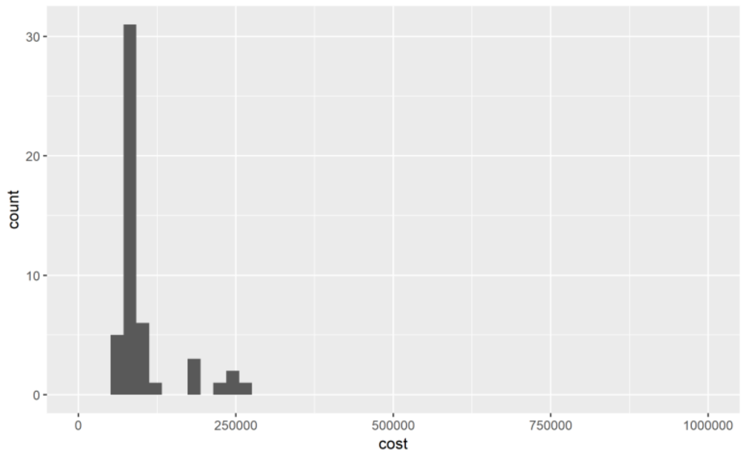


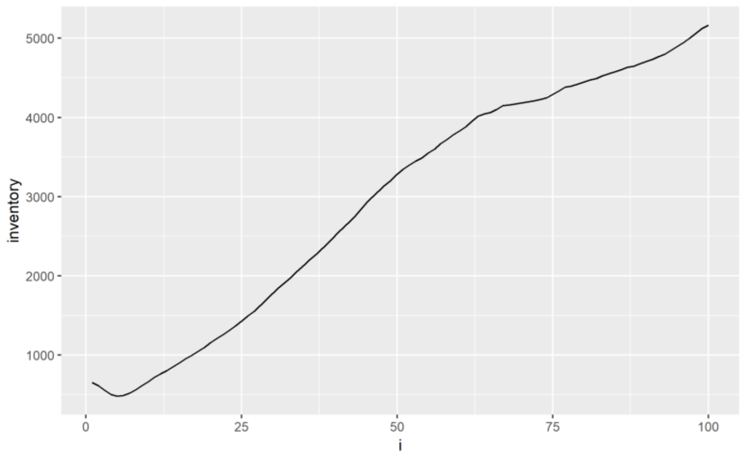
300 iterations NN





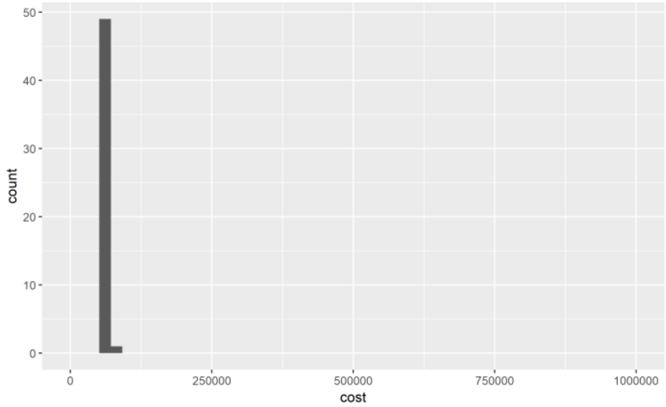
325 iterations NN

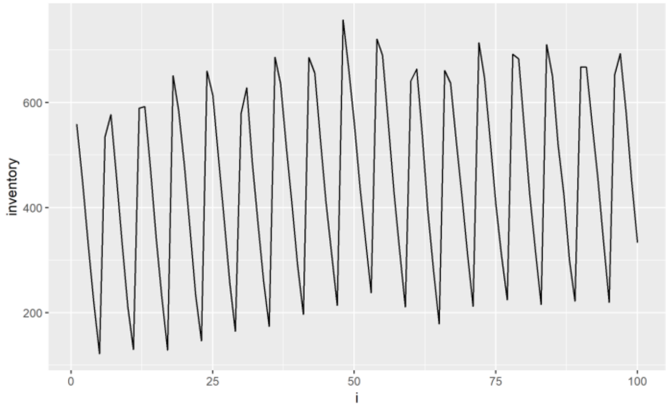




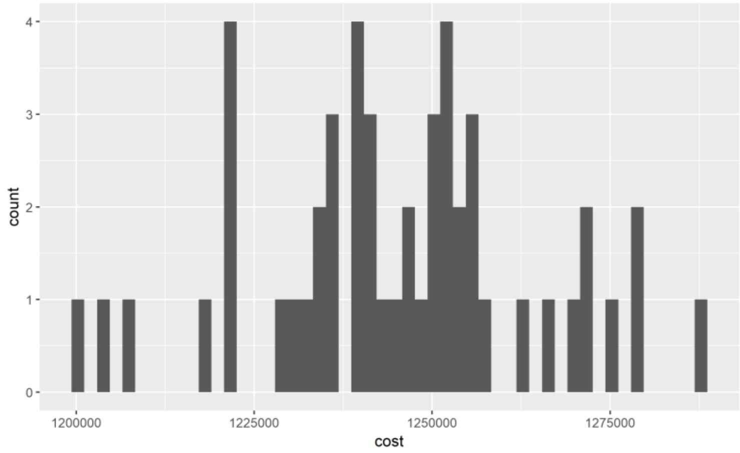
Trial 3: var_rel = 1 / delivery_ontime = 0,85

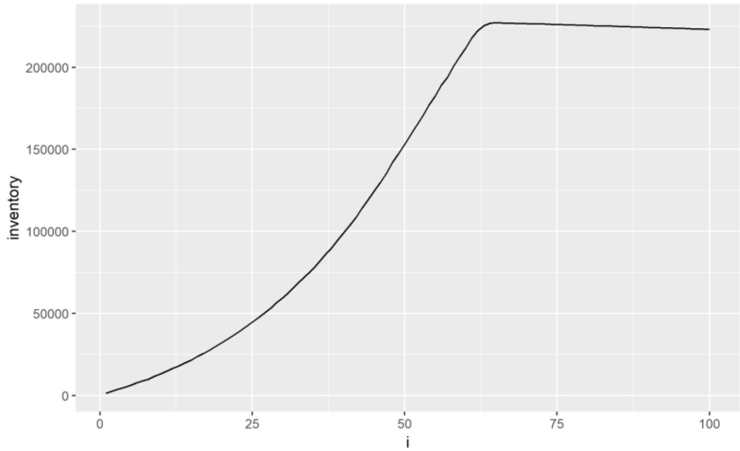
50 EOQ experiments



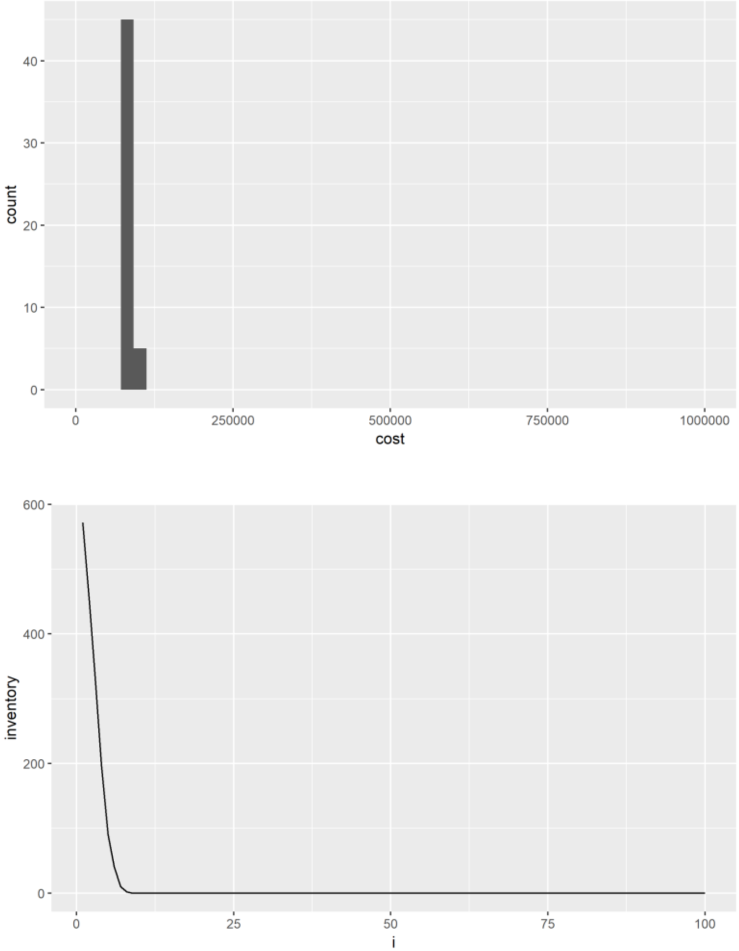


50 iterations NN

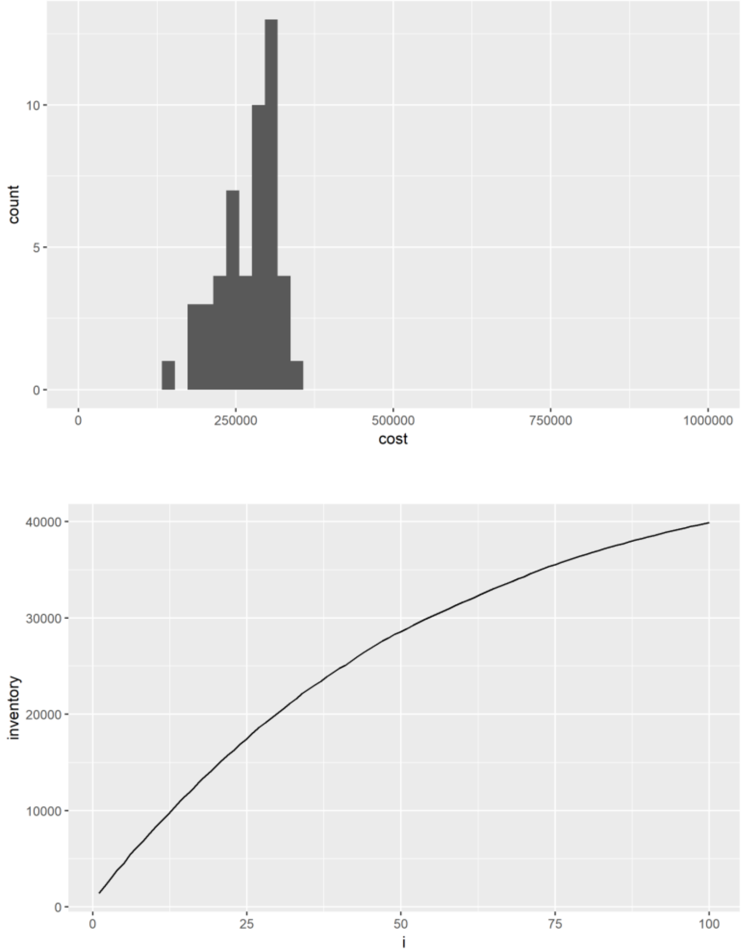




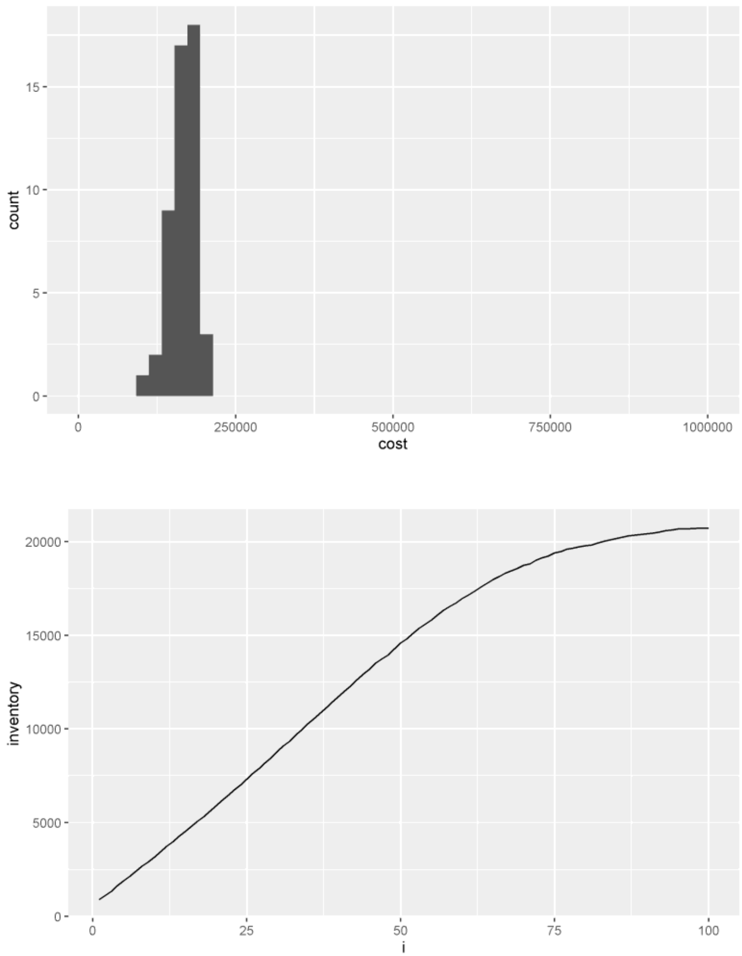
75 iterations NN



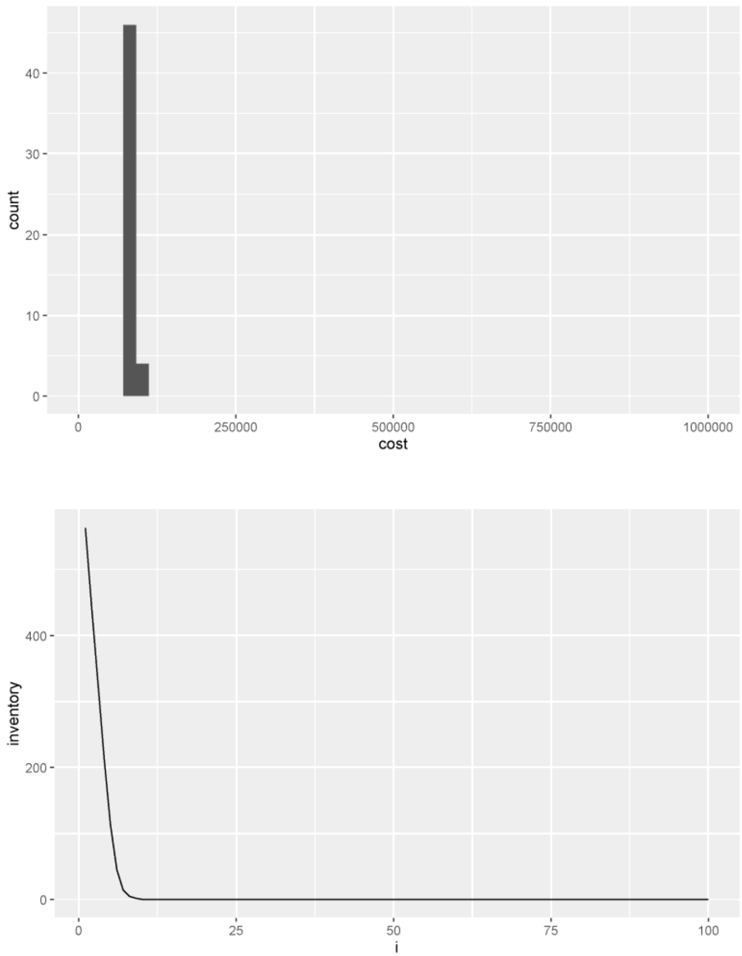
100 iterations NN



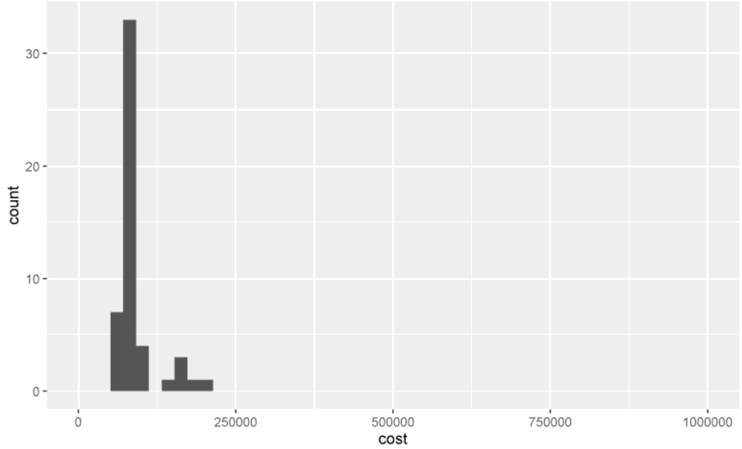
125 iterations NN

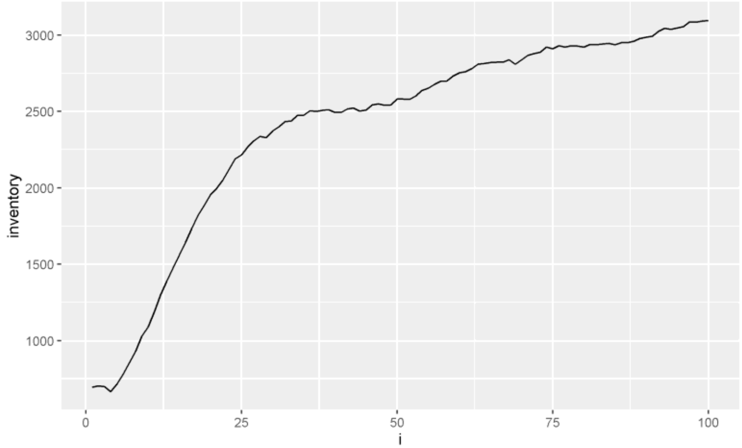


150 iterations NN

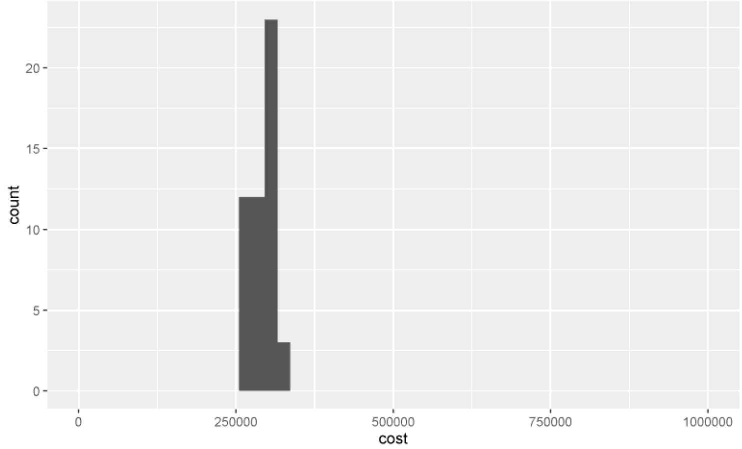


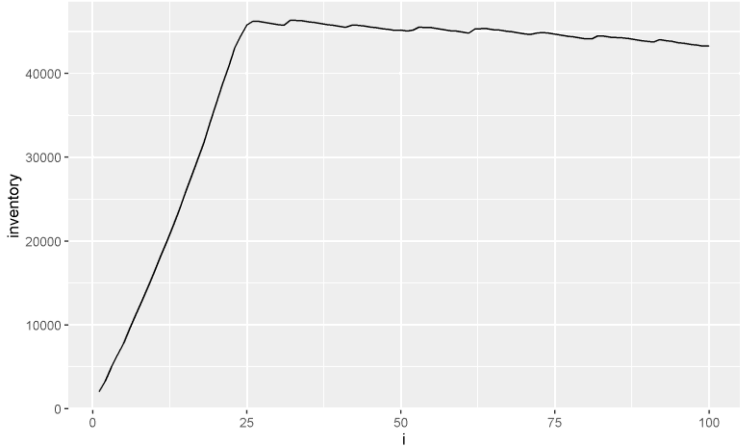
175 iterations NN



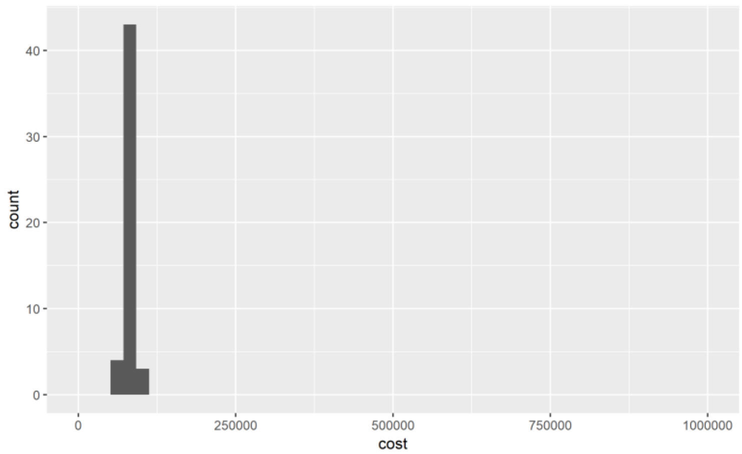


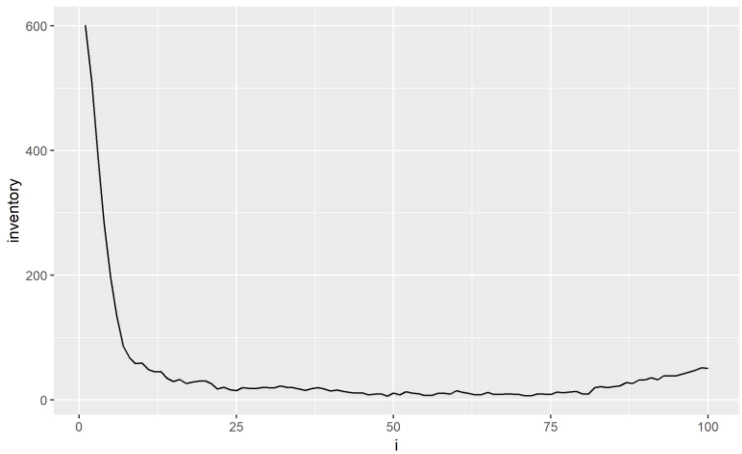
200 iterations NN



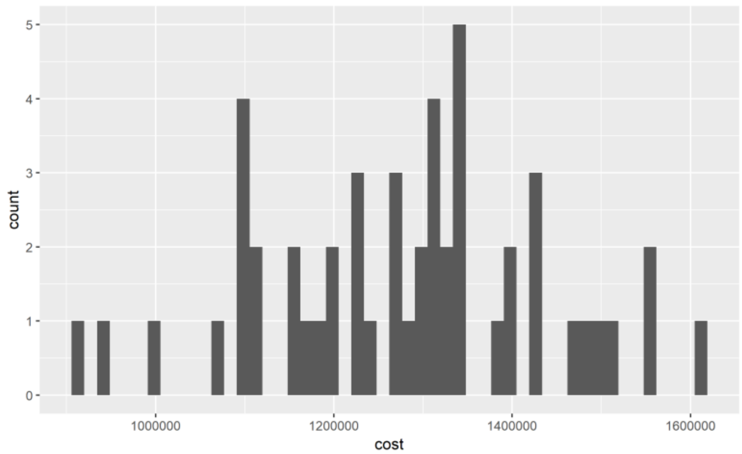


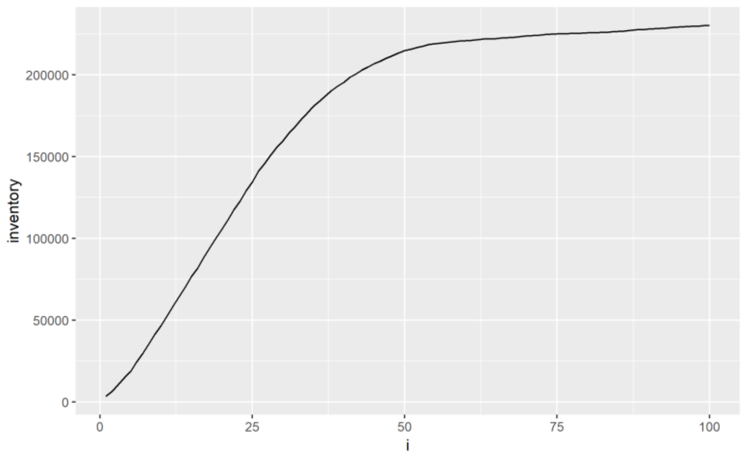
225 iterations NN



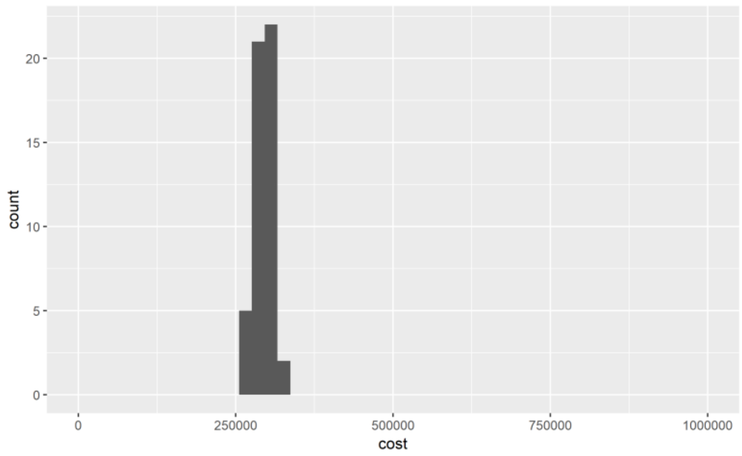


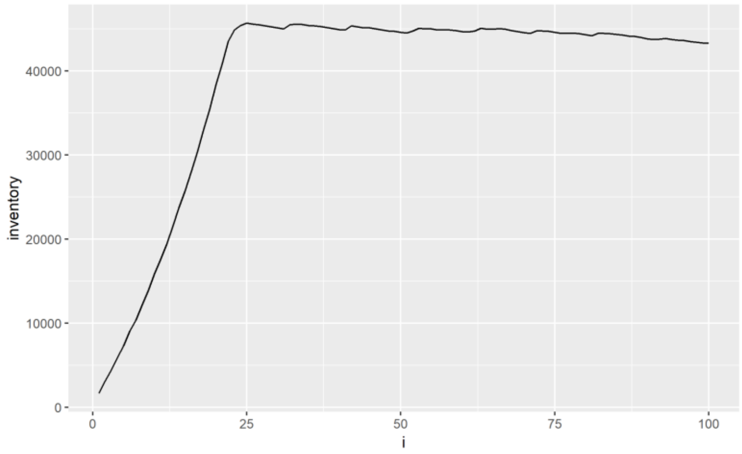
250 iterations NN



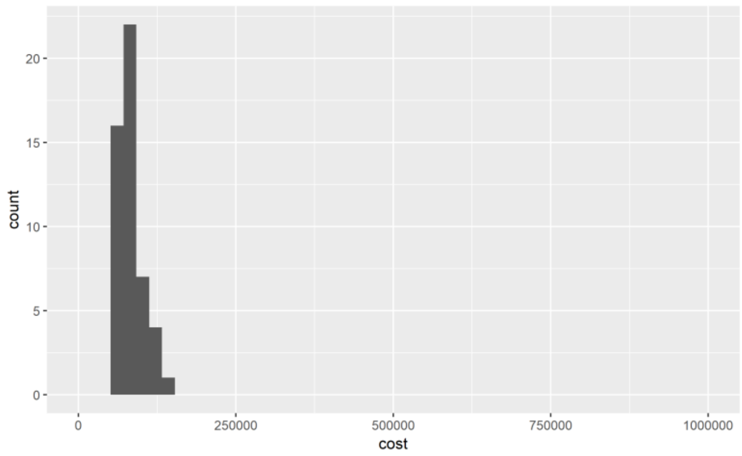


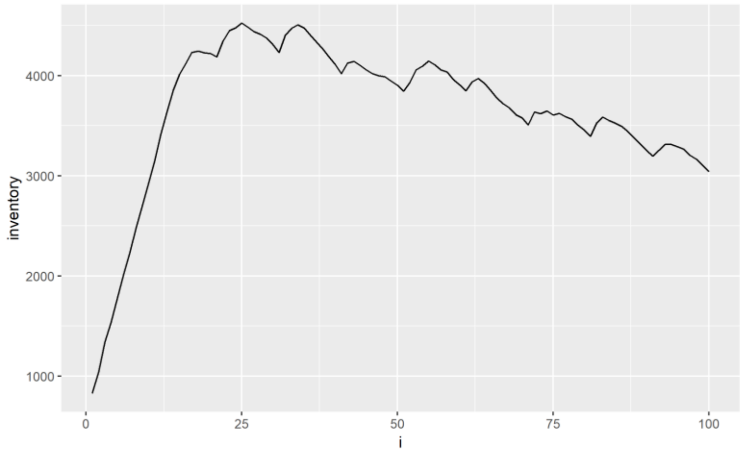
275 iterations NN



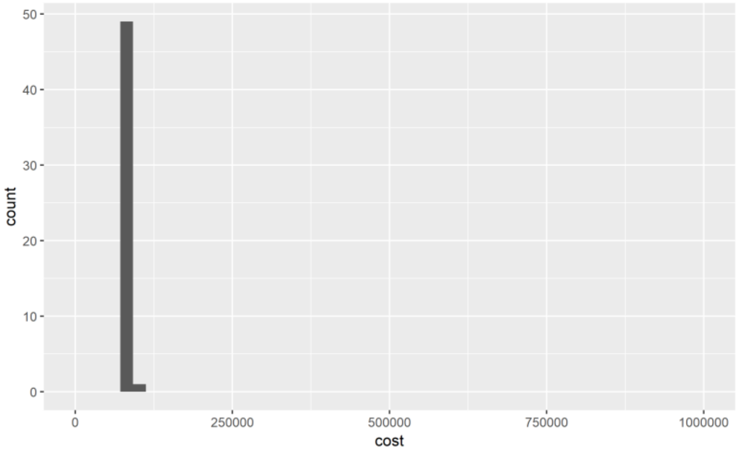


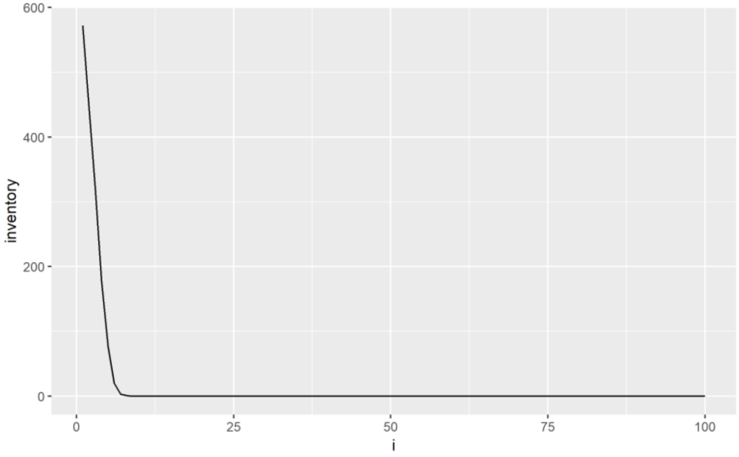
300 iterations NN





325 iterations NN





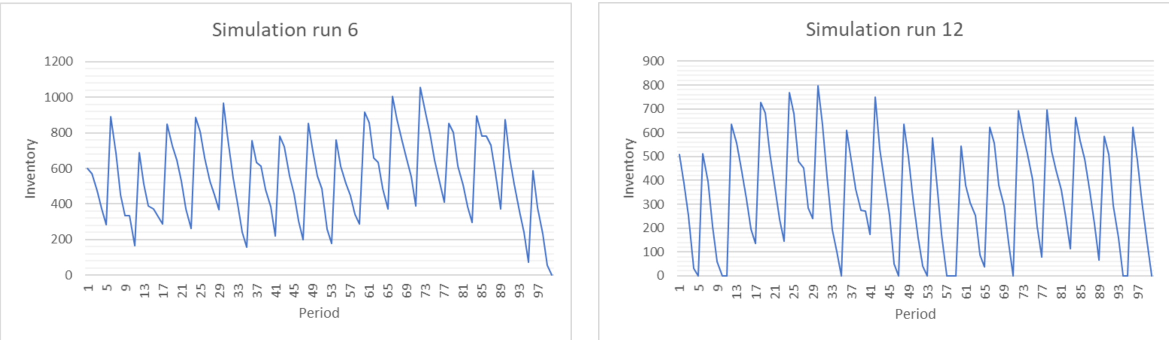
C. Analysis of the results

Trial 1: var_rel = 1 / delivery_ontime = 1

EOQ experiments

	sim_run	c_supply	c_ordering	c_lost_sales	c_inventory	Total costs
Lowest costs	6	55440	160	105	270,59	55975,59
		99,04%	0,29%	0,19%	0,48%	
Highest costs	12	55440	160	10672,5	166,215	66438,72
		83,45%	0,24%	16,06%	0,25%	
Mean	All					60104,97

C 1. Comparison of costs between simulation run 6 and simulation run 12



C 2. Inventory level graphs for simulation run 6 and simulation run 12

325 iterations NN

simulation_run	c_supply	c_ordering	c_lost_sales	inventory	Total costs
6	38290	1000	21900	32,35	61222,35
11	54695	1000	7680	201,425	63576,425
12	55605	1000	7582,5	221,35	64408,85
13	53780	1000	7935	187,6	62902,6
18	52720	1000	15472,5	161,385	69353,885
22	56865	1000	0	996,275	58861,275
30	48285	690	11355	1241,255	61571,255
31	17065	820	51847,5	15,46	69747,96
32	44315	1000	14632,5	78,335	60025,835
33	31405	840	38865	35,08	71145,08
35	46325	910	21142,5	244,825	68622,325
37	41520	1000	22950	89,975	65559,975
38	43690	990	25710	310,45	70700,45
49	38225	1000	24195	29,615	63449,615
50	61000	1000	3577,5	472,63	66050,13

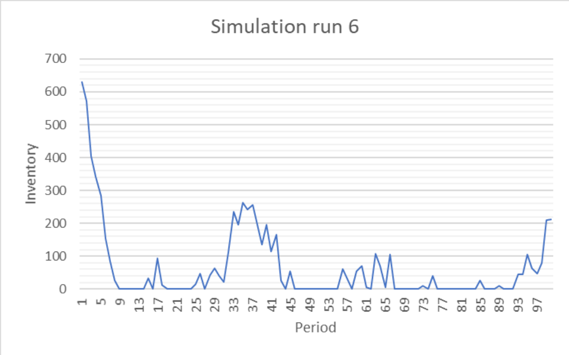
C 3. Costs for the 15 simulation runs with the lowest costs

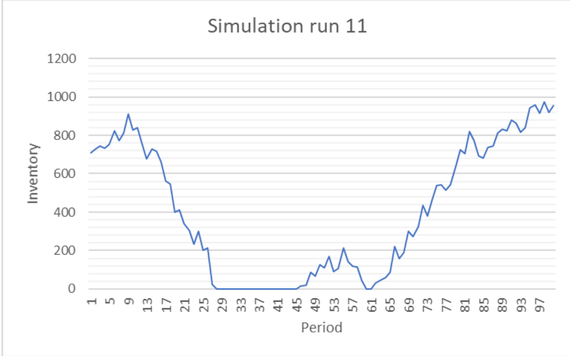
simulation_run	c_supply	c_ordering	c_lost_sales	inventory
6	62,54%	1,63%	35,77%	0,05%
11	86,03%	1,57%	12,08%	0,32%
12	86,33%	1,55%	11,77%	0,34%
13	85,50%	1,59%	12,61%	0,30%
18	76,02%	1,44%	22,31%	0,23%
22	96,61%	1,70%	0,00%	1,69%
30	78,42%	1,12%	18,44%	2,02%
31	24,47%	1,18%	74,34%	0,02%
32	73,83%	1,67%	24,38%	0,13%
33	44,14%	1,18%	54,63%	0,05%
35	67,51%	1,33%	30,81%	0,36%
37	63,33%	1,53%	35,01%	0,14%
38	61,80%	1,40%	36,36%	0,44%
49	60,24%	1,58%	38,13%	0,05%
50	92,35%	1,51%	5,42%	0,72%
Mean	70,61%	1,46%	27,47%	0,46%

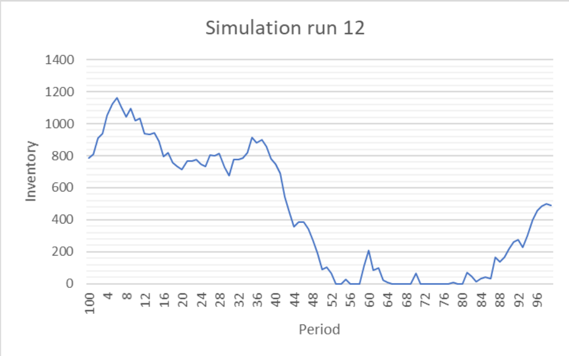
C 4. Distribution of costs for the 15 simulation runs with the lowest costs

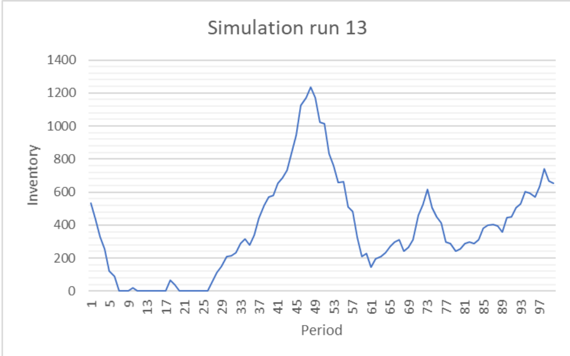
simulation_run	MIN	MAX	MEAN
6	7	142	76,58
11	37	138	109,39
12	53	166	111,21
13	17	182	107,56
18	33	174	105,44
22	43	187	113,73
30	0	285	96,57
31	0	129	34,13
32	42	144	88,63
33	0	137	62,81
35	0	226	92,65
37	14	132	83,04
38	0	207	87,38
49	9	134	76,45
50	17	237	122

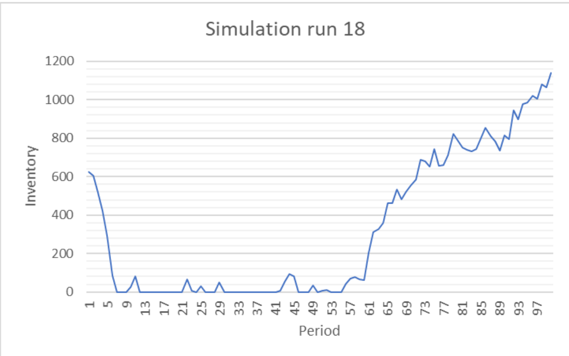
C 5. Variability of items ordered for the 15 simulation runs with the lowest costs

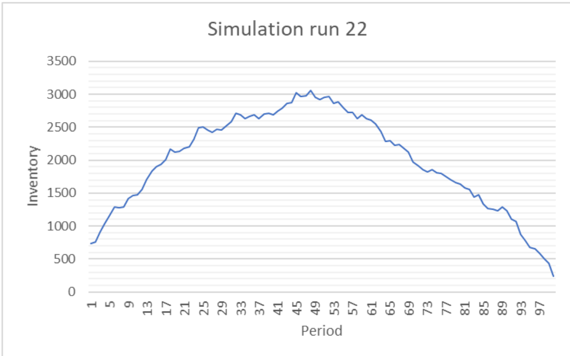

Simulation run 6
Inventory
700 600 500 400 300 200 100 0
1 5 9 13 17 21 25 29 33 37 41 45 49 53 57 61 65 69 73 77 81 85 89 93 97
Period

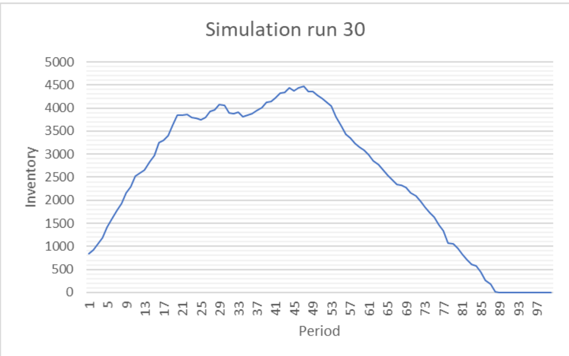

Simulation run 11
Inventory
1200 1000 800 600 400 200 0
1 5 9 13 17 21 25 29 33 37 41 45 49 53 57 61 65 69 73 77 81 85 89 93 97
Period

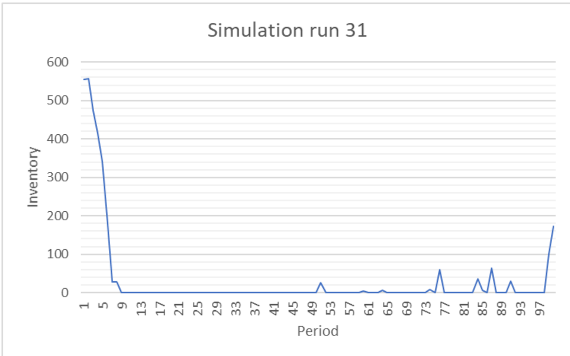

Simulation run 12
Inventory
1400 1200 1000 800 600 400 200 0
100 4 8 12 16 20 24 28 32 36 40 44 48 52 56 60 64 68 72 76 80 84 88 92 96
Period

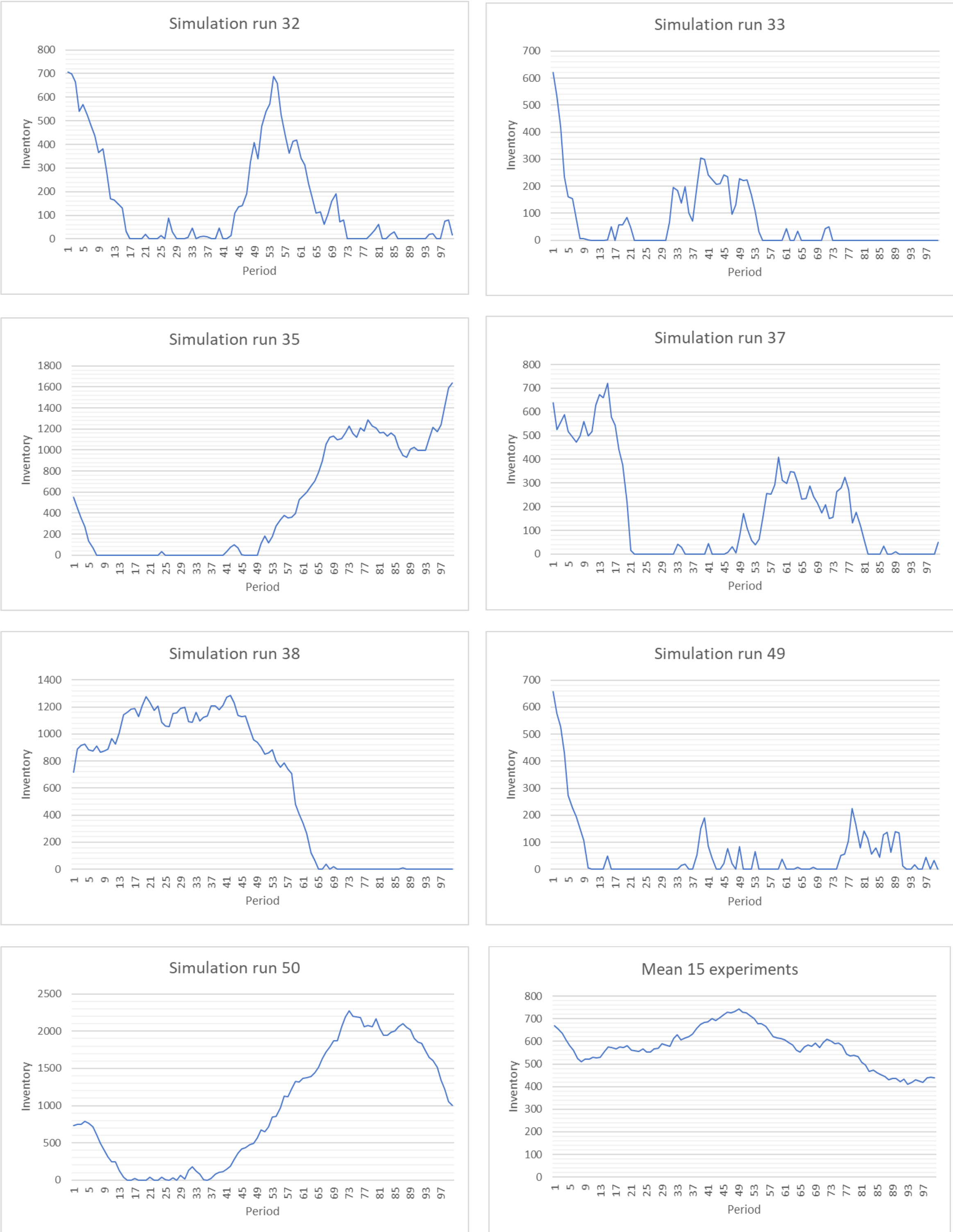

Simulation run 13
Inventory
1400 1200 1000 800 600 400 200 0
1 5 9 13 17 21 25 29 33 37 41 45 49 53 57 61 65 69 73 77 81 85 89 93 97
Period


Simulation run 18
Inventory
1200 1000 800 600 400 200 0
1 5 9 13 17 21 25 29 33 37 41 45 49 53 57 61 65 69 73 77 81 85 89 93 97
Period


Simulation run 22
Inventory
3500 3000 2500 2000 1500 1000 500 0
1 5 9 13 17 21 25 29 33 37 41 45 49 53 57 61 65 69 73 77 81 85 89 93 97
Period


Simulation run 30
Inventory
5000 4500 4000 3500 3000 2500 2000 1500 1000 500 0
1 5 9 13 17 21 25 29 33 37 41 45 49 53 57 61 65 69 73 77 81 85 89 93 97
Period


Simulation run 31
Inventory
600 500 400 300 200 100 0
1 5 9 13 17 21 25 29 33 37 41 45 49 53 57 61 65 69 73 77 81 85 89 93 97
Period



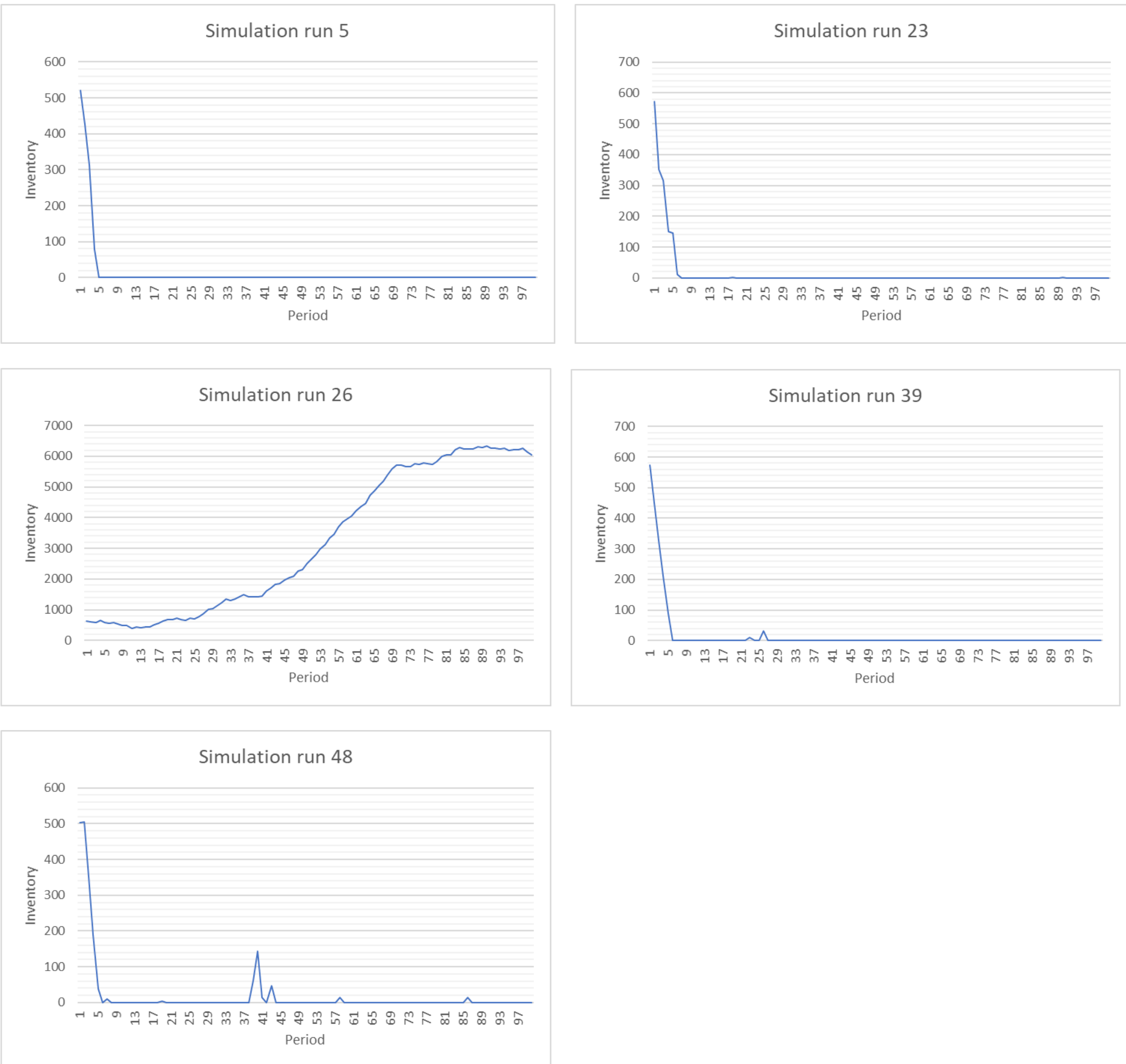
C 6. Inventory level graphs for the 15 simulation runs with the lowest costs

simulation_run	c_supply	c_ordering	c_lost_sales	inventory	Total costs
5	0	0	91387,5	6,69	91394,19
23	8305	760	78472,5	7,77	87545,27
26	89325	1000	0	1593,43	91918,43
39	9335	550	81930	8,475	91823,475
48	6590	580	82717,5	9,455	89896,955

C 7. Costs for the 5 simulation runs with the highest costs

simulation_run	c_supply	c_ordering	c_lost_sales	inventory
5	0,00%	0,00%	99,99%	0,01%
23	9,49%	0,87%	89,64%	0,01%
26	97,18%	1,09%	0,00%	1,73%
39	10,17%	0,60%	89,23%	0,01%
48	7,33%	0,65%	92,01%	0,01%

C 8. Distribution of costs for the 5 simulation runs with the highest costs



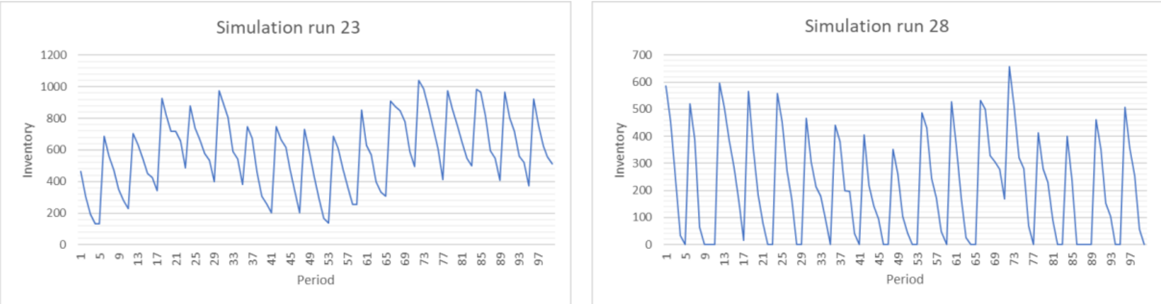
C 9. Inventory level graphs for the 5 simulation runs with the highest costs

Trial 2: var_rel = 0,85 - 1 / delivery_ontime = 1

EOQ experiments

	sim_run	c_supply	c_ordering	c_lost_sales	c_inventory	Total costs
Lowest costs	23	50745	160	0	290,22	51195,22
		99,12%	0,31%	0,00%	0,57%	
Highest costs	28	50475	160	18622,5	106,615	69364,12
		98,59%	0,31%	36,38%	0,21%	
Mean	All					60626,37

C 10. Comparison of costs between simulation run 23 and simulation run 28



C 11. Inventory level graphs for simulation run 23 and simulation run 28

	sim_run 23	sim_run 28	All
ordered	11088	11088	554400
delivered	10149	10095	513209
% reliability	91,53%	91,04%	92,57%

C 12. Comparison % delivery reliability between simulation run 23 and simulation run 28

325 iterations NN

simulation_run	c_supply	c_ordering	c_lost_sales	inventory	Total costs
5	38845	680	26002,5	100,31	65627,81
14	68210	550	0	2422,38	71182,38
26	32925	510	35887,5	97,945	69420,445
28	47840	380	9270	1340,855	58830,855
47	52080	470	7695	1394,945	61639,945

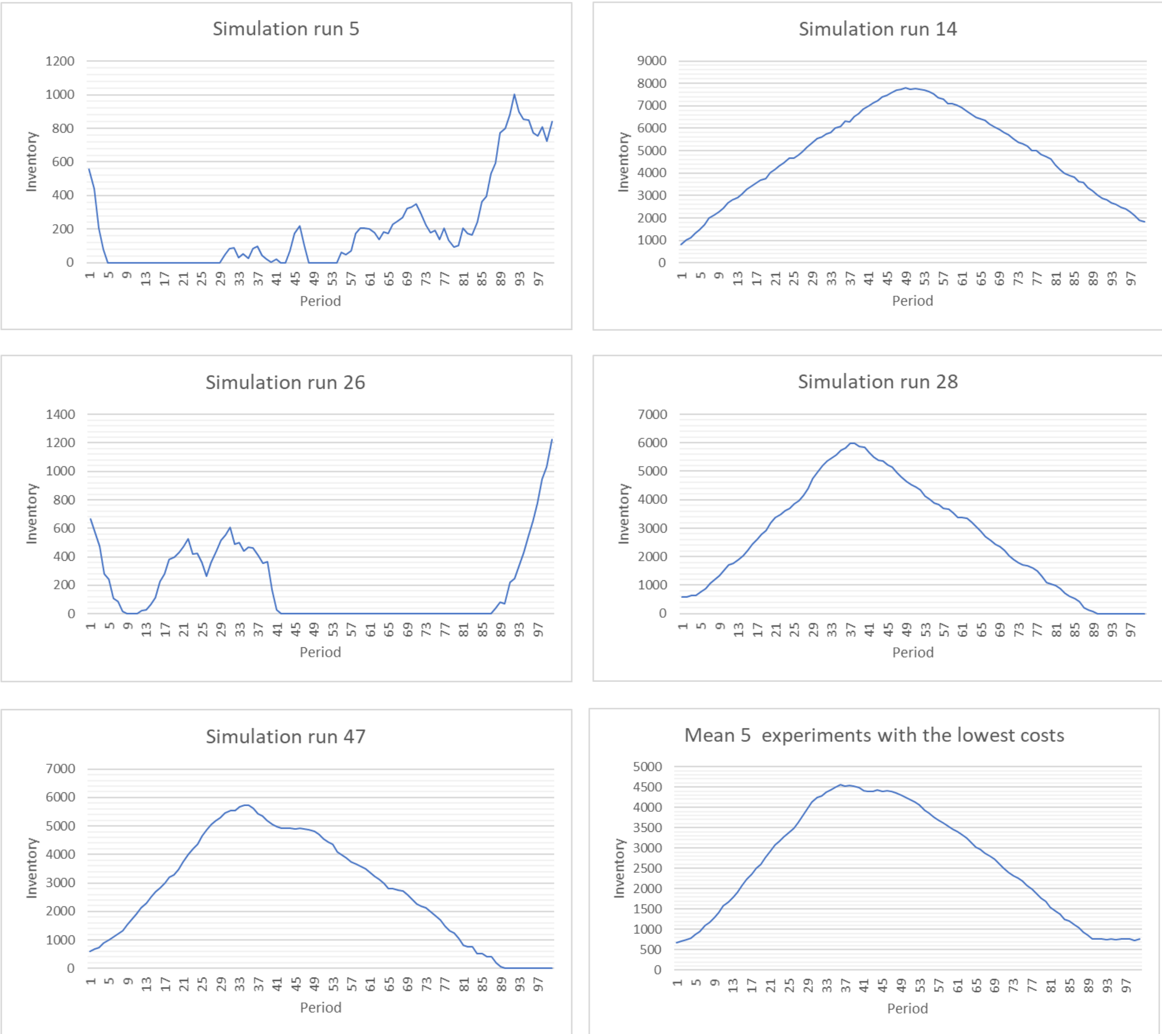
C 13. Costs for the 5 simulation runs with the lowest costs

simulation_run	c_supply	c_ordering	c_lost_sales	inventory
5	59,19%	1,04%	39,62%	0,15%
14	95,82%	0,77%	0,00%	3,40%
26	47,43%	0,73%	51,70%	0,14%
28	81,32%	0,65%	15,76%	2,28%
47	84,49%	0,76%	12,48%	2,26%

C 14. Distribution of costs for the 5 simulation runs with the lowest costs

sim_run	% reliability
5	93,40%
14	92,19%
26	92,98%
28	92,61%
47	92,85%
Mean	92,81%

C 15. Comparison of the % delivery reliability between the 5 simulation runs with the lowest costs



C 16. Inventory level graphs for the 5 simulation runs with the lowest costs

simulation_run	c_supply	c_ordering	c_lost_sales	inventory	Total costs
17	227440	510	0	13709,88	241659,88
23	259190	990	0	6218,8	266398,8
41	0	0	87607,5	8,955	87616,455
42	173580	510	0	9650,69	183740,69
50	211095	630	0	10691,67	222416,67

C 17. Costs for the 5 simulation runs with the highest costs

simulation_run	c_supply	c_ordering	c_lost_sales	inventory
17	94,12%	0,21%	0,00%	5,67%
23	97,29%	0,37%	0,00%	2,33%
41	0,00%	0,00%	99,99%	0,01%
42	94,47%	0,28%	0,00%	5,25%
50	94,91%	0,28%	0,00%	4,81%

C 18. Distribution of costs for the 5 simulation runs with the highest costs

sim_run	% reliability
17	92,23%
23	92,78%
41	-
42	93,00%
50	93,08%
Mean	92,77%

C 19. Comparison of the % delivery reliability between the 5 simulation runs with the highest costs

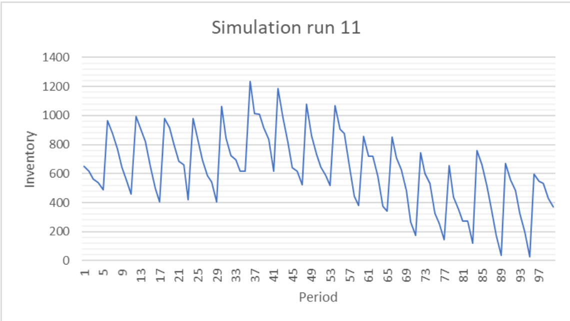
C 20. Inventory level graphs for the 5 simulation runs with the highest costs

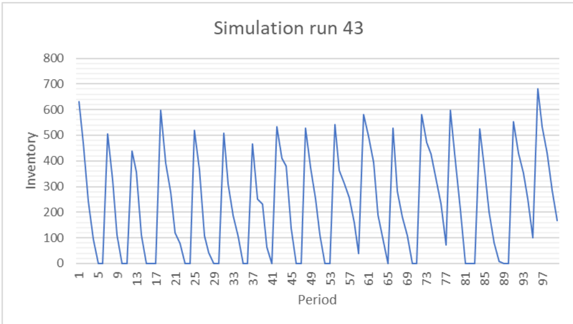
Trial 3: var_rel = 1 / delivery_ontime = 0,85

EOQ experiments

	sim_run	c_supply	c_ordering	c_lost_sales	c_inventory	Total costs
Lowest costs	11	55440	160	0	311,34	55911,34
		99,16%	0,29%	0,00%	0,56%	
Highest costs	43	55440	160	19177,5	117,135	74894,64
		74,02%	0,21%	25,61%	0,16%	
Mean						60637,38

C 21. Comparison of costs between simulation run 11 and simulation run 43





C 22. Inventory level graphs for simulation run 11 and simulation run 43

Simulation run 43		
Late?	c_lost_sales	%
Yes	6045	31,52%
No	13132,5	68,48%
Total	19177,5	100,00%

C 23. Lost sales cost due to delivery delay for simualtion 43

All experiments		
Late?	c_lost_sales	%
Yes	48240	20,05%
No	192405	79,95%
Total	240645	100,00%

C 24. Lost sales cost due to delivery delay for all experiments

All experiments		
sales	572415	94,69%
lostsales	32086	5,31%
Total	604501	100,00%

C 25. Number of lost sales and sales for all experiments

300 iterations NN

simulation_run	c_supply	c_ordering	c_lost_sales	inventory	Total costs
1	39120	260	18487,5	871,925	58739,425
7	68765	320	0	2057,645	71142,645
10	41330	300	18142,5	832,74	60605,24
12	29370	290	35280	493,105	65433,105
18	52245	270	3862,5	1574,91	57952,41
20	60985	310	2235	1375,04	64905,04
23	51970	360	4252,5	966,715	57549,215
29	53815	470	0	1219,08	55504,08
32	65945	340	0	1934,04	68219,04
35	64735	370	0	1792,96	66897,96
38	22075	210	44662,5	336,74	67284,24
39	36375	190	32685	810,29	70060,29
43	60365	370	0	1405,515	62140,515
17	33480	180	37770	617,64	72047,64
27	71770	400	0	2106,48	74276,48
44	71765	420	0	1824,26	74009,26

C 26. Costs for the 16 simulation runs with the lowest costs

simulation_run	c_supply	c_ordering	c_lost_sales	inventory
1	66,60%	0,44%	31,47%	1,48%
7	96,66%	0,45%	0,00%	2,89%
10	68,20%	0,50%	29,94%	1,37%
12	44,89%	0,44%	53,92%	0,75%
18	90,15%	0,47%	6,66%	2,72%
20	93,96%	0,48%	3,44%	2,12%
23	90,31%	0,63%	7,39%	1,68%
29	96,96%	0,85%	0,00%	2,20%
32	96,67%	0,50%	0,00%	2,84%
35	96,77%	0,55%	0,00%	2,68%
38	32,81%	0,31%	66,38%	0,50%
39	51,92%	0,27%	46,65%	1,16%
43	97,14%	0,60%	0,00%	2,26%
17	46,47%	0,25%	52,42%	0,86%
27	96,63%	0,54%	0,00%	2,84%
44	96,97%	0,57%	0,00%	2,46%
Mean	78,94%	0,49%	18,64%	1,93%

C 27. Distribution of costs for the 16 simulation runs with the lowest costs

All experiments	
% periods one_day_late	Number of experiments
1%	5
2%	5
3%	7
4%	4
5%	3
6%	5
7%	4
8%	2
9%	2
Total	37
%	74,00%

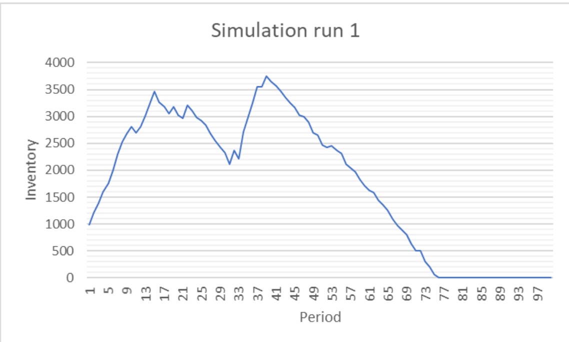
C 28. Number of experiments that are affected by a delay and to what extent

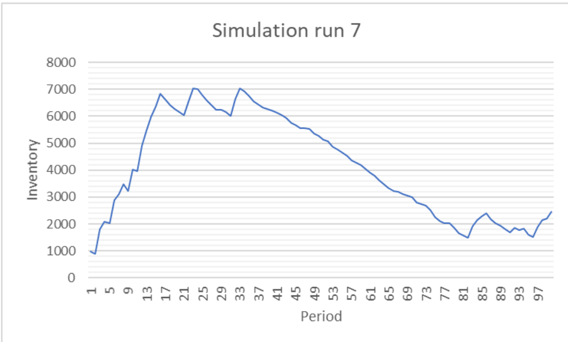
16 experiments with lowest costs	
% periods one_day_late	Number of experiments
1%	2
2%	3
3%	5
4%	2
5%	0
6%	2
7%	1
8%	0
9%	0
Total	15
%	93,75%

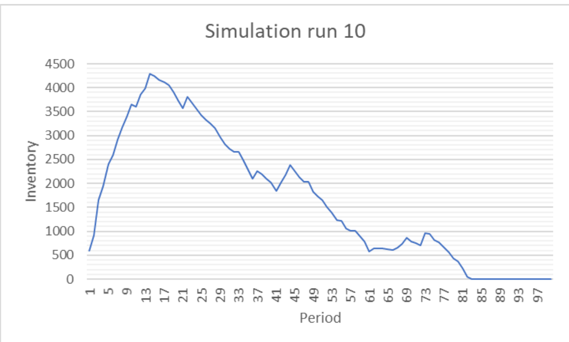
C 29. Number of experiments that are affected by a delay and to what extent

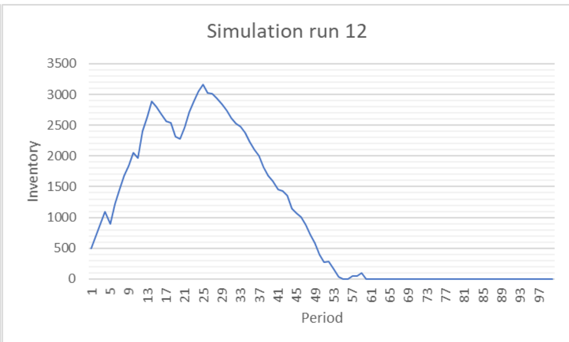
16 simulation runs with the lowest costs		
Late?	c_lost_sales	%
Yes	2460	1,25%
No	194917,5	98,75%
Total	197377,5	100,00%

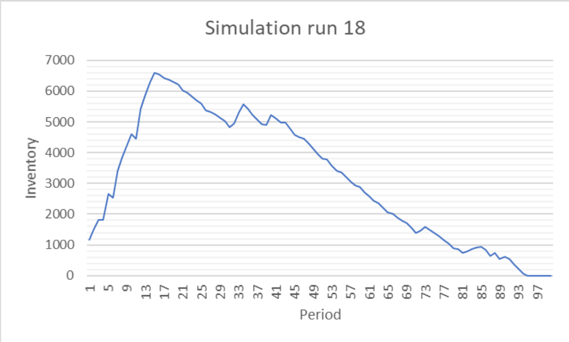
C 30. Lost sales cost due to delivery delay for simualtion 43

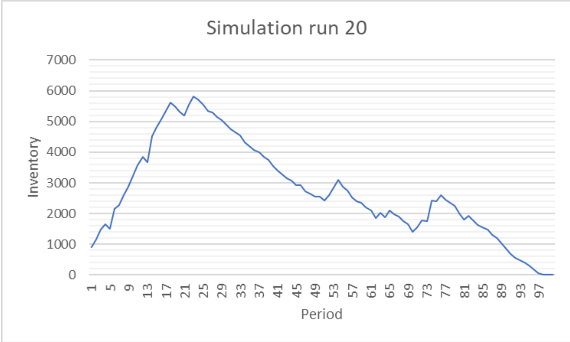


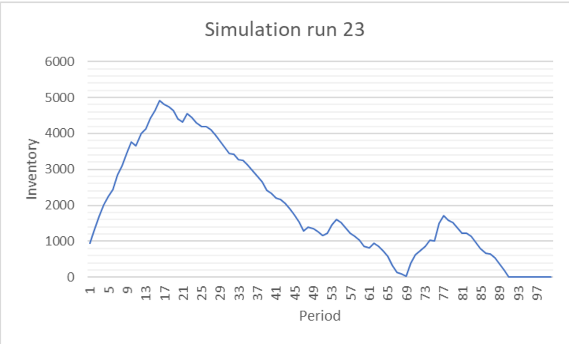


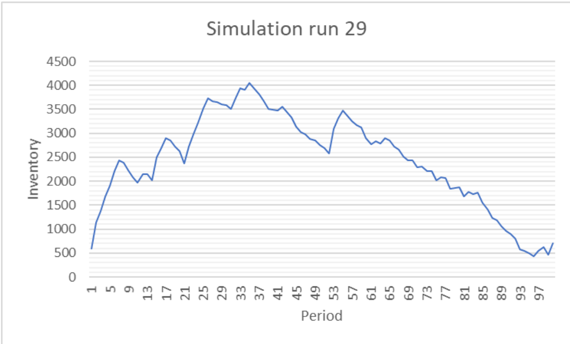


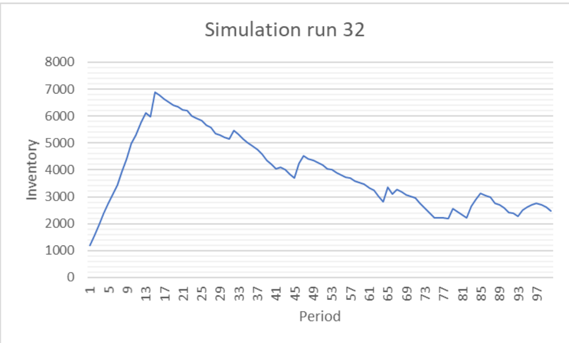


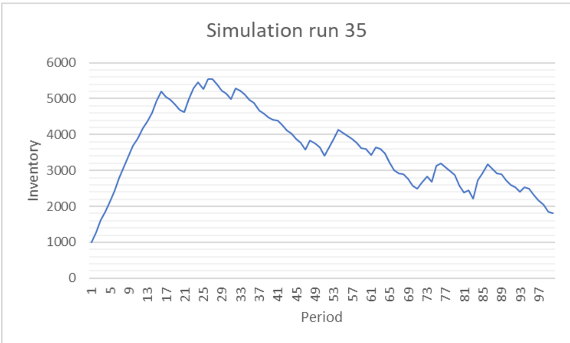

Simulation run 18
Inventory
Period

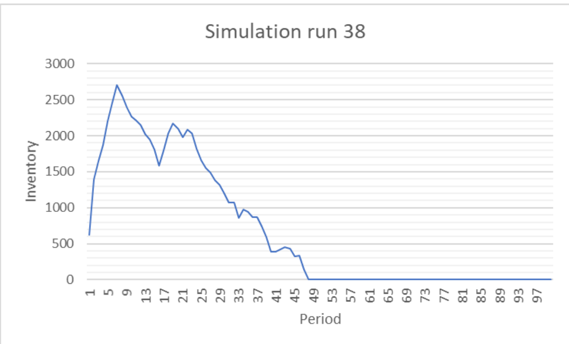

Simulation run 20
Inventory
Period

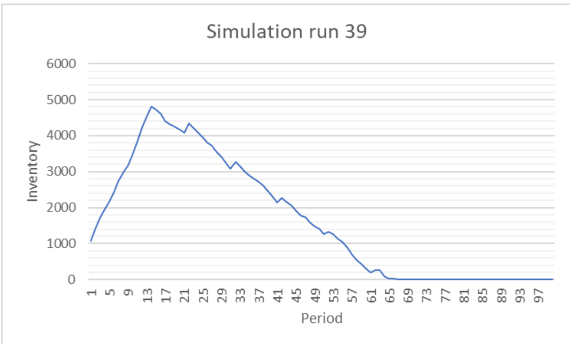

Simulation run 23
Inventory
Period

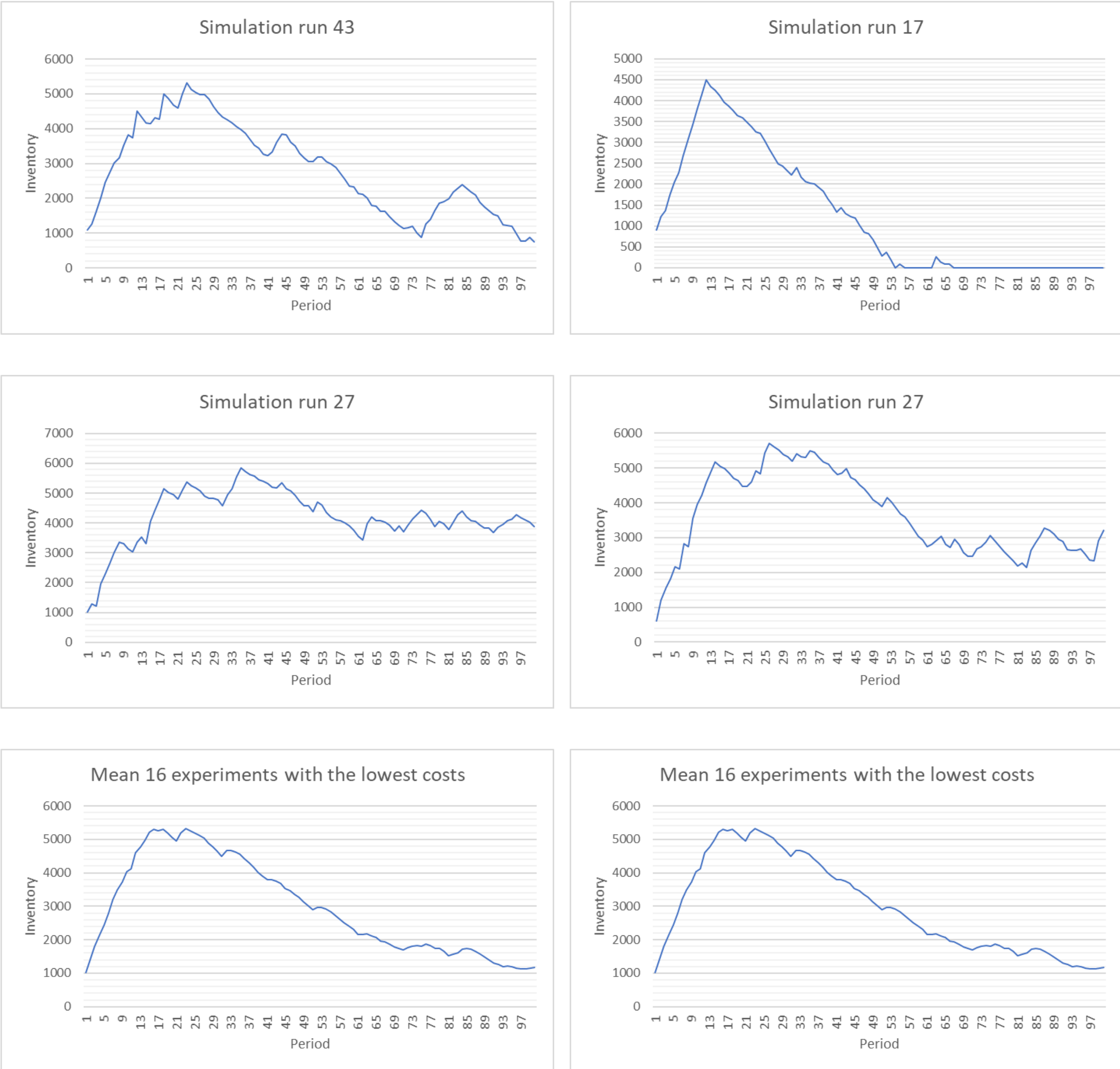

Simulation run 29
Inventory
Period


Simulation run 32
Inventory
Period


Simulation run 35
Inventory
Period


Simulation run 38
Inventory
Period


Simulation run 39
Inventory
Period



C 31. Inventory level graphs for the 5 simulation runs with the highest costs

5 experiments with the highest costs	
% periods one_day_late	**Number of experiments**
1%	0
2%	0
3%	0
4%	0
5%	1
6%	2
7%	0
8%	1
9%	1
Total	5
%	100,00%

C 32. Number of experiments that are affected by a delay and to what extent

simulation_run	c_supply	c_ordering	c_lost_sales	inventory	Total costs
9	119475	320	0	5078,165	124873,17
16	115880	320	0	4645,71	120845,71
19	114865	310	0	4935,32	120110,32
30	118815	420	0	4637,8	123872,8
46	120570	300	0	5529,44	126399,44

C 33. Costs for the 5 simulation runs with the highest costs

simulation_run	c_supply	c_ordering	c_lost_sales	inventory
9	95,68%	0,26%	0,00%	4,07%
16	95,89%	0,26%	0,00%	3,84%
19	95,63%	0,26%	0,00%	4,11%
30	95,92%	0,34%	0,00%	3,74%
46	95,39%	0,24%	0,00%	4,37%

C 34. Distribution of costs for the 5 simulation runs with the highest costs

C 35. Distribution of costs for the 5 simulation runs with the highest costs